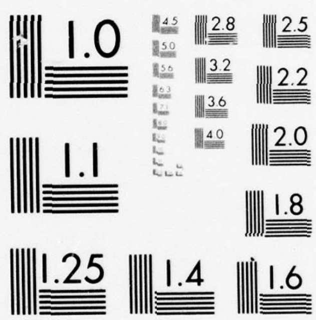
AD-A052 143 VIRGINIA UNIV CHARLOTTESVILLE RESEARCH LABS FOR THE--ETC F/G 8/2
FORMULATION OF A SPACE OBLIQUE MERCATOR MAP PROJECTION.(U)
FEB 78 J L JUNKINS, J D TURNER DAAG53-76-C-0067
UNCLASSIFIED UVA/525023/ESS77/105 ETL-0131 NL

1 OF 2
AD
A052143



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ETL-0131

$\checkmark$ G366870

(12) 2

AD A052143

AD No. ____
DDC FILE COPY

RESEARCH LABORATORIES FOR THE ENGINEERING SCIENCES

SCHOOL OF ENGINEERING AND APPLIED SCIENCE

University of Virginia

Charlottesville, Virginia 22901

Final Report

FORMULATION OF A SPACE OBLIQUE MERCATOR MAP PROJECTION

Phase III, Contract No. DAAG-53-76-C-0067

Submitted to:

Computer Science Laboratory
Code 82000
U.S. Army Engineer Topographic Laboratories
Fort Belvoir, Virginia 22060

Submitted by:

John L. Junkins
Associate Professor

James D. Turner
Research Assistant

Report No. UVA/525023/ESS77/105

November 1977

Revised February 1978

RESEARCH LABORATORIES FOR THE ENGINEERING SCIENCES

Members of the faculty who teach at the undergraduate and graduate levels and a number of professional engineers and scientists whose primary activity is research generate and conduct the investigations that make up the school's research program. The School of Engineering and Applied Science of the University of Virginia believes that research goes hand in hand with teaching. Early in the development of its graduate training program, the School recognized that men and women engaged in research should be as free as possible of the administrative duties involved in sponsored research. In 1959, therefore, the Research Laboratories for the Engineering Sciences (RLES) was established and assigned the administrative responsibility for such research within the School.

The director of RLES—himself a faculty member and researcher—maintains familiarity with the support requirements of the research under way. He is aided by an Academic Advisory Committee made up of a faculty representative from each academic department of the School. This Committee serves to inform RLES of the needs and perspectives of the research program.

In addition to administrative support, RLES is charged with providing certain technical assistance. Because it is not practical for each department to become self-sufficient in all phases of the supporting technology essential to present-day research, RLES makes services available through the following support groups: Machine Shop, Instrumentation, Facilities Services, Publications (including photographic facilities), and Computer Terminal Maintenance.

9 Final Report on Phase 3,

6 FORMULATION OF A SPACE OBLIQUE MERCATOR MAP PROJECTION.

Phase III, Contract No. 15 DAAG-53-76-C-0067

Submitted to:

Computer Science Laboratory
Code 82000
U.S. Army Engineer Topographic Laboratories
Fort Belvoir, Virginia 22060

Submitted by:

10 John L. Junkins James D. Turner
Associate Professor

James D. Turner
Research Assistant

11 Feb 78 18 ETL

12 130P. 19 0131

Department of Mechanical and Aerospace Engineering
RESEARCH LABORATORIES FOR THE ENGINEERING SCIENCES
SCHOOL OF ENGINEERING AND APPLIED SCIENCE
UNIVERSITY OF VIRGINIA
CHARLOTTESVILLE, VIRGINIA

Report No. 14 UVA/525023/ESS77/105 Copy No._____

November 1977

Revised February 1978

366 870 JOB

ABSTRACT

This report develops a *dynamic* map projection especially suited for processing and display of satellite electro-optical remote sensing of the earth's surface. The new map projection (the Space Oblique Mercator) projects the satellite ground-track from the ellipsoid into the map plane, free of length distortion and free of *normal view curvature* distortion. The length and curvature distortions in the finite sensed region are negligible for most applications. The report details the formulation, provides numerical examples for the LANDSAT multi-spectral scanner, and includes FORTRAN IV software as an appendix.

TABLE OF CONTENTS

TABLE OF CONTENTS (cont'd.)

1.0 SUMMARY

We report herein upon a research and development effort which has led to the following results:

- A rigorous mathematical development of a Space Oblique Mercator map projection. This projection is especially designed for processing and displaying data from LANDSAT-type satellite-borne multi-spectral scanners and is characterized by the following desirable features:
  - Zero length and angle distortion along the satellite groundtrack, and only a few parts in 10,000 length distortion as far as 200 km off typical groundtracks.
  - Rigorously valid for an *arbitrary* continuous satellite orbit; the formulation can be routinely interfaced to state-of-the-art orbit integration programs, or can use simplified Keplerian or circular orbit approximations, depending upon the needs of particular applications.
  - Rigorously valid for an arbitrary ellipsoid reference figure for the earth.
  - Computationally efficient; the most expensive calculations are orbit-dependent integrals which need only be determined once (for each specific nominal orbit).
- Prototype software (FORTRAN IV) has been developed, checked out, and is demonstrated herein. The software has been developed and all calculations performed on the University of Virginia's CYBER 172 computer system.

2.0 PREFACE

This report constitutes the final report of Phase III of contract no. DAAG-53-76-C-0067 performed by the University of Virginia for the U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia, under the sponsorship of the U.S. Geological Survey, (USGS), Reston, Virginia.

The authors acknowledge the competent guidance and technical liasion of Mr. L.A. Gambino (technical mointor, USAETL) and Dr. A.P. Colvocoresses (EROS Program Cartography Coordinator, USGS).

3.0 INTRODUCTION

In Ref. 1, Colvocoresses conceived a *dynamic* map projection concept especially suited to satellite mapping. Unlike classical *static* map projections, the line which is projected distortion-free is not restricted to be an equator, a meridian, a parallel, or an oblique great circle; rather, the distortion-free line is the satellite sub-point path (groundtrack) on a reference ellipsoid. Colvocoresses developed a geometrical analog involving an oscillating cylinder to which projections are made from the reference ellipsoid, the cylinder oscillation is such that the cylinder instantaneously osculates with the normal sub-point on the ellipsoid. A small region near the sub-point, when projected from the ellipsoid onto the oscillating cylinder and developed onto a plane, is projected with negligible length and angle distortions.

Unfortunately, Colvocoresses' elegant geometric analog was not supported by a mathematical formulation of the map projection equations; rather, he issued a challenge "for the cartographic community to undertake a considerable and dedicated effort to develop the mathematical model and associated computer programs" to implement this map projection concept. The present report documents our response to the above challenge. While motivated (and occasionally perplexed!) by the oscillating cylinder analog, we have not used this concept in our formulations. We elected instead to set down the mathematical constraints underlying Colvocoresses' objectives. These lead immediately to differential equations which are the key to a rigorous solution to the problem.

We have successfully developed a most general version of this space oblique mercator (SOM) map projection formulation (really, an infinite family of map projections, depending upon definition of the nominal orbit). The derivation, computational summary, and software for the SOM projection are documented herein.

The present report is arranged in a modular fashion. The computational summary concentrates on the structure of the solution and relegates details to appendixes. Exclusive of Section 6, the report is directed primarily toward readers seeking to understand the essence of and learn to use the map projection. Section 6 is intended to document the geometrical, mathematical, and intuitive details and to develop the equations in a fashion which parallels their invention.

4.0 SOM COMPUTATIONAL SUMMARY

4.1 Forward Transformation

The map coordinates (x,y) are related to the corresponding ellipsoidal coordinates (ϕ,λ) by the formulae

$$x = \int_0^{t^*} V\cos f \, dt + R_c \ln[\tan(\frac{\pi}{4} + \frac{\alpha}{2})] \cos \gamma_s/\cos\theta_s \qquad (1a)$$

$$y = \int_0^{t^*} V\sin f \, dt + R_c \ln[\tan(\frac{\pi}{4} + \frac{\alpha}{2})] \sin \gamma_s/\cos\theta_s + y_g(0) \qquad (1b)$$

where

t = time since some selected initial point in the orbit

t^*= the "scan instant" for which the scan vector passes through (ϕ,λ).

V = satellite sub-point's instantaneous speed relative to the earth

ρ = the instantaneous radius of curvature of the sub-point path, as projected into the plane tangent to the ellipsoid at the sub-point.

$f \equiv \int^t \frac{V}{\rho} d\tau$: τ = dummy (time) integration variable. (2)

R_c= the local radius of curvature of the ellipsoid in a plane whose normal is the earth-fixed velocity vector of the sub-point

$(\alpha,\gamma_s,\theta_s)$ angles defined in §6.3.

Methods for evaluating the right-hand side of equations (1) are now discussed.

4.1.1 Calculation of the Integrals $\int_0^t V\cos f \, d\tau$, $\int_0^t V\sin f \, d\tau$, $\int_0^t \frac{V}{\rho} d\tau$

As will be seen in §6.1, the integral terms of equations (1a) and (1b) are correctly interpreted as (x_g,y_g), the map coordinates of the groundtrack or sub-point path. We can write instead of the three integrals, the three differential equations

$$\dot{x}_g \equiv \frac{dx_g}{dt} = V\cos f \tag{3a}$$

$$\dot{y}_g \equiv \frac{dy_g}{dt} = V\sin f \tag{3b}$$

$$\dot{f} \equiv \frac{df}{dt} = \frac{V}{\rho} \tag{3c}$$

Given the appropriate formulas (Appendixes A and B) for calculation of V and ρ, the given initial conditions $\{x_g(o),\ y_g(o),\ f(o)\}$, then equations (3) can be integrated numerically to evaluate the integrals in equations (1) and (2). However, it is expensive to carry out these integrations many times, and since these integrals have been found to be very smooth functions of time, they can be conveniently and accurately replaced by their harmonic series as

$$x_g(t) = \bar{\dot{x}}_g\ t + \sum_{n=0}^{\infty} \{S_{x_g n} \sin(\frac{n\pi t}{P}) + C_{x_g n} \cos(\frac{n\pi t}{P})\} \equiv \int_o^t V \cos f\ d\tau \tag{4a}$$

$$y_g(t) = \bar{\dot{y}}_g\ t + \sum_{n=0}^{\infty} \{S_{y_g n} \sin(\frac{n\pi t}{P}) + C_{y_g n} \cos(\frac{n\pi t}{P})\} \equiv \int_o^t V \sin f\ d\tau \tag{4b}$$

$$f(t) = \bar{\dot{f}}\ t + \sum_{n=0}^{\infty} \{S_{fn} \sin(\frac{n\pi t}{P}) + C_{fn} \cos(\frac{n\pi t}{P})\} \equiv \int_o^t \frac{V}{\rho}\ d\tau \tag{4c}$$

where

P = orbital period

$$\bar{\dot{x}}_g = \text{the average value of } \frac{dx_g}{dt} = \frac{1}{P}\int_o^P \frac{dx_g}{d\tau}(\tau)\ d\tau,\ x_g \to y_g, f\ ^* \tag{5}$$

---

*The symbol "→" (interpreted as "replaced by") allows the three equations

$$\bar{\dot{x}}_g = \frac{1}{P}\int_o^P (\frac{dx_g}{dt})\ d\tau$$

$$\bar{\dot{y}}_g = \frac{1}{P}\int_o^P (\frac{dy_g}{dt}) d\tau$$

$$\bar{\dot{f}} = \frac{1}{P}\int_o^P (\frac{df}{dt}) d\tau$$

(footnote continued on next page)

$$S_{x_{gn}} = \frac{2}{P} \int_0^P [x_g(\tau) - \bar{\dot{x}}_g \tau] \sin\left(\frac{n\pi\tau}{P}\right) d\tau \quad , \; x_g \to y_g, f \tag{6}$$

$$C_{x_{gn}} = \frac{2}{P} \int_0^P [x_g(\tau) - \bar{\dot{x}}_g \tau] \cos\left(\frac{n\pi\tau}{P}\right) d\tau, \quad x_g \to y_g, f \tag{7}$$

For certain approximate orbits and choice of the initial point (position in the orbit for which t=0), many of the above coefficients are either zero or negligible. For example, for circular LANDSAT orbits, (99° inclination, 103 min. period) equations (4) have been found to reduce to simply

$$x_g(t) = \bar{\dot{x}}_g t + A_1 \sin\left(\frac{\pi t}{P}\right) + A_2 \sin\left(\frac{2\pi t}{P}\right) + \; . \; . \; . \; + A_9 \sin\left(\frac{9\pi t}{P}\right) \tag{8a}$$

$$y_g(t) = B_o + B_1 \cos\left(\frac{\pi t}{P}\right) + B_2 \sin\left(\frac{2\pi t}{P}\right) + \; . \; . \; . \; . \; + B_9 \sin\left(\frac{9\pi t}{P}\right) \tag{8b}$$

$$f(t) = C_1 \sin\left(\frac{\pi t}{P}\right) + C_2 \sin\left(\frac{2\pi t}{P}\right) + \; . \; . \; . \; + C_9 \sin\left(\frac{9\pi t}{P}\right) \tag{8c}$$

where the specialized version of equations (5), (6), and (7) are

$$\bar{\dot{x}}_g = \frac{1}{P} \int_o^P \frac{dx_g}{d\tau} \, d\tau \tag{9a}$$

$$A_n = \frac{2}{P} \int_o^P [x_g(\tau) - \bar{\dot{x}}_g \, \tau] \sin\left(\frac{n\pi\tau}{P}\right) d\tau \tag{9b}$$

$$= \frac{2}{P} \int_o^P y_g(\tau) \, \cos\left(\frac{n\pi\tau}{P}\right) d\tau \tag{9c}$$

$$C_n = \frac{2}{P} \int_o^P f(\tau) \, \sin\left(\frac{n\pi\tau}{P}\right) d\tau \tag{9d}$$

and with t=0 the instant for which the satellite is at its northernmost lattitude. In eqs. 8, all but two coefficients are near-negligible in each of the three series (see §5.1).

The solution procedure (restricting the discussion to the LANDSAT case) for the coefficients (9) is as follows:

- Using the Runge-Kutta algorithm of Appendix B and the calculations leading to instantaneous values for V and f (established in Appendix B), calculate $\bar{\dot{x}}_g$ by numerical integration of the differential

---

*footnote cont'd.: to be written as only the first, eq. (5), since replacing x_g by y_g yields the second and replacing x_g by f yields the third. This shorthand notations is used in all subsequent equations for compactness.

equation

$$\frac{d}{dt}(\bar{\dot{x}}_g) = (\frac{1}{P}) V\cos f \tag{10}$$

using zero as the initial condition.

- The integrals (9) are evaluated by simultaneous Runge-Kutta solution (Appendix C) of the following system of (3+3n) differential equations

$$\frac{df}{dt} = \frac{V}{\rho} \tag{11a}$$

$$\frac{dx_g}{dt} = V\cos f \tag{11b}$$

$$\frac{dy_g}{dt} = V\sin f \tag{11c}$$

$$\frac{dA_n}{dt} = \frac{2}{P}[x_g(t) - \bar{\dot{x}}_g\, t]\sin(\frac{n\pi t}{P}) \tag{11d}$$

$$\frac{dB_n}{dt} = \frac{2}{P} y_g(t) \cos(\frac{n\pi t}{P}) \tag{11e}$$

$$\frac{dC_n}{dt} = \frac{2}{P} f(t) \sin(\frac{n\pi t}{P}) \tag{11f}$$

using zeros for initial values.

Analogous integrations establish the coefficients for the general case of equations (5), (6), and (7).

4.1.2 Scan Time Determination

For given ellipsoidal coordinates (ϕ,λ), the corresponding scan time t^* is defined as the instant that the plane established by the unit vector $\hat{\underline{n}}$ (normal to the ellipsoid) and the scan vector $\hat{\underline{w}}$ (tangent plane projection of the orbit normal) contain the vector $\Delta\underline{R}$ [from the sub-point

(ϕ_g, λ_g) to the point of interest (ϕ,λ) on the ellipsoid]; this geometrical condition requires that these three vectors satisfy the constraint

$$F(t^*) = [(\underline{\hat{w}} \times \underline{\hat{n}}) \cdot \Delta\underline{R}]_{t=t^*} = 0$$
$$= [\Delta R_x(\hat{w}_y\hat{n}_z - \hat{w}_z\hat{n}_y) + \Delta R_y(\hat{w}_z\hat{n}_x - \hat{w}_x\hat{n}_z) + \Delta R_z(\hat{w}_x\hat{n}_y - \hat{w}_y\hat{n}_x)]_{t=t^*} \tag{12}$$

This constraint neglects the finite scanner sweep time; upon finding a t^* satisfying (12), an additive correction (13b) is introduced to account for the scanner sweep time.

In Appendix B, the explicit earth-fixed components of $\underline{\hat{w}}$ and $\underline{\hat{n}}$ are given, and the components of $\Delta\underline{R}$ are developed in §6.3; these expressions (6.13) and their derivatives are required to calculate $F(t)$ and $\frac{dF(t)}{dt}$. The time t^* for which (12) vanishes is determined via a Newton's successive approximation algorithm as

$$t^{(k+1)} = t^{(k)} - \frac{F[t^{(k)}]}{\left.\frac{dF}{dt}\right|_{t=t^{(k)}}} \tag{13a}$$

with the approximate estimates $t^{(o)}$ determined by calculating the angle $\Delta\theta$ between the initial sub-point position vector and the position vector to (ϕ,λ), then dividing by $2\pi/P$: $t^{(o)} = \frac{\Delta\theta}{2\pi/P}$, with appropriate quadrant checks.

The condition (12) and the resulting converged t^* from (13a) must be corrected to account for the finite scan time Δt for the scanner to sweep from the center of scan $[t(\phi_g,\lambda_g)]$out to the sensed point at (ϕ,λ). This correction, added to (13a), has the form

$$\Delta t = T\left(\frac{\varepsilon}{\varepsilon_{max}}\right) \tag{13b}$$

where

$T = \frac{1}{2}$ the scan period

$= 18.355$ milliseconds (for the LANDSAT scanner)

ε_{max} = the maximum scanner beam deflection angle away from local vertical

$= 5.78$ degrees (for the LANDSAT scanner)

ε = instantaneous scanner beam deflection angle away from the local vertical

$$= \cos^{-1} \left(\frac{-\underline{S} \cdot \underline{H}}{SH}\right) \cdot \text{sign} \, (\Delta \underline{R} \cdot \hat{\underline{c}}) \qquad (13c)$$

where from figures 6.4C, 6.4D, the displacement of the sensed point from the satellite is,

$$\underline{S} = \underline{R} - \underline{r}$$

Equation (13b) assumes a constant linear scan rate and that center-of-scan is exactly on the sub-point path; these idealizations should be replaced if more precise vehicle altitude and scanner dynamics are available.

4.1.3 Calculation of SOM Map Coordinates

Upon convergence (usually 4 or 5 iterations) of eqn. (13a), with t* the converged value, the integral terms of the map projection equations (1) can be evaluated from the series (8a) and (8b). The second terms of equations (1) are then calculated immediately, given the values of α, γ_s, and θ_s from Appendix B for the instant t=t*.

4.2 Inverse SOM Transformation

The ellipsoidal coordinates (ϕ, λ) corresponding to map coordinates (x,y) are determined as follows:

First the instant t^* that the scan vector $\hat{\underline{w}}$ is colinear with the displacement vector $\delta\underline{R}$ from the groundtrack (x_g, y_g) to the point of interest (x,y) is determined via the iteration

$$t^{(k+1)} = t^{(k)} - \frac{G[t^{(k)}]}{\left.\frac{dG}{dt}\right|_{t=t^{(k)}}} \tag{14}$$

where

$$G(t) = (\hat{\underline{w}} \times \hat{\underline{n}}) \cdot \delta\underline{R} \tag{15}$$

$$\delta\underline{R} = [x - x_g(t)]\,\underline{i} + [y - y_g(t)]\,\underline{j} \tag{16}$$

Explicit equations are obtained for calculation of $G(t)$ and $\frac{dG(t)}{dt}$ by substituting $\hat{\underline{n}}$ and $\hat{\underline{w}}$ from Appendix B. The starting approximation for (14) is taken as $t^{(o)} \cong x/\bar{x}_g$. The converged value [usually requiring 3 or 4 iterations of (14)] is t^*. The unit vectors $\underline{i}$ and $\underline{j}$ are parallel to the x and y axis of the mapping plane. The correction for finite scan time, eqn. (13b), should be added to the converged t^* from (14). The associated partials assume an infinite scan speed.

Knowing t^*, it is a simple matter to determine (ϕ,λ) via the 2 variable successive approximation

$$\begin{Bmatrix} \phi^{(k+1)} \\ \lambda^{(k+1)} \end{Bmatrix} = \begin{Bmatrix} \phi^{(k)} \\ \lambda^{(k)} \end{Bmatrix} + \begin{bmatrix} \left.\frac{\partial x}{\partial \phi}\right|^{(k)} & \left.\frac{\partial x}{\partial \lambda}\right|^{(k)} \\ \left.\frac{\partial y}{\partial \phi}\right|^{(k)} & \left.\frac{\partial y}{\partial \lambda}\right|^{(k)} \end{bmatrix}^{-1} \begin{Bmatrix} x - x^{(k)} \\ y - y^{(k)} \end{Bmatrix} \tag{17}$$

where $\phi^{(k)}$ and $\lambda^{(k)}$ are substituted into eqns. (1) to determine $x^{(k)}$, $y^{(k)}$; and the partial derivative matrix is determined via the formulas summarized in §6.4. The iteration (17) has been found very well behaved (usually converging in 4 to 5 iterations). The starting estimates are taken as $\phi^{(o)} = \phi_g(t^*)$, $\lambda^{(o)} = \lambda_g(t^*)$, calculated from formulas of Appendix A.

5.0 COMPUTATIONAL TESTS FOR THE ERTS-1 (LANDSAT) ORBIT

We adopted a nominal orbit with an inclination of 99°, a period of 103.267 minutes, and zero eccentricity. This orbit approximates that of the Earth Resources Technology Satellite (ERTS-1, also known as LANDSAT). For the reference ellipsoid, we adopted the values

a = equatorial radius = 6378.165 km

b = polar radius = 6356.783 km

These correspond to a flattening of 1/298.3 or an eccentricity of 0.0818130. In earth-fixed coordinates, this satellite generates the sub-point trace shown in the oblique views of figures 5.1a and 5.1b. For the special case of the LANDSAT orbit, we summarize below numerical results of the major options of the software documented in Appendix D.

5.1 LANDSAT Groundtrack Projection

Following the approach of 4.1.1, the software of Appendix D was used to evaluate the groundtrack's projection into the map plane as

$$x_g(t) = \int_0^t V\cos f\, d\tau = A_o t + \sum_{n=1}^{9} A_n \sin\left(\frac{n\pi t}{P}\right) \tag{5.1a}$$

$$y_g(t) = \int_0^t V\sin f\, d\tau + y_g(t_o) = \sum_{n=1}^{9} B_n \cos\left(\frac{n\pi t}{P}\right) \tag{5.1b}$$

where the coefficients were found to be

$$A_o \equiv \bar{x}_g \equiv \frac{1}{P}\int_0^P \frac{dx_g}{dt}\, dt = 6.504961 \text{ km/sec} \tag{5.2a}$$

$$A_n = \frac{2}{P}\int^P [x_g(t) - \bar{x}_g\, t]\sin\left(\frac{n\pi t}{P}\right)dt \tag{5.2b}$$

$$B_n = \frac{2}{P}\int^P y_g(t)\cos\left(\frac{n\pi t}{P}\right)dt \tag{5.2c}$$

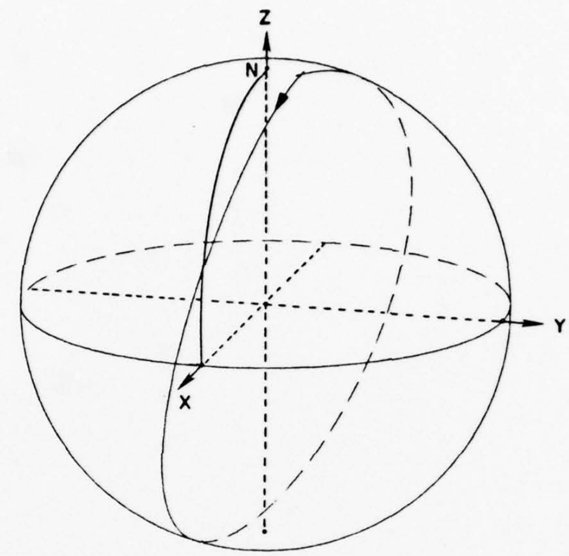


Figure 5.1 A

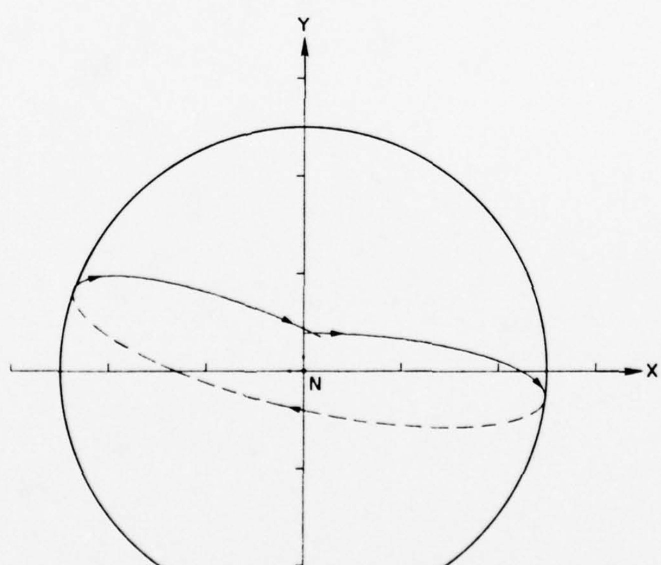


Figure 5.1 B

Figure 5.1 LANDSAT Subpoint Trajectory (Groundtrack) on the Earth-Fixed Reference Ellipsoid

n	A_n	B
0	6.50496	
1	+.00000	-.00000
2	-.00000	916.45110
3	-.00000	.00000
4	9.73648	-.00000
5	.00000	.00000
6	.00000	-.09648
7	-.00000	.00000
8	.00232	-.00000
9	-.00000	.00000

As is evident from the coefficients, only two harmonic terms each are actually required. This fact is evident only <u>after</u> the coefficients are determined and in general is a function of the particular orbit.

Appendix D provides Table D1*listing ϕ_g, λ_g, x_g, y_g, for 100 equally spaced time increments spanning the orbit. The map plane projection (x_g,y_g) of the LANDSAT groundtrack is the bold line of figure 5.2.

5.2 Example Forward and Inverse Transformations

Using the LANDSAT orbit data and ref. ellipsoid of §5.0 and the calculation sequence summarized in §4.1, the following ellipsoid point

$$(\phi,\lambda) = (-0^{o}.16549 \quad , \quad -7^{o}.02310)$$

resulted [eqns.(1), using software of Appendix D] in the following map plane coordinates

$$(x,y) = (10099.66 \text{ km}, - 58.587 \text{ km}).$$

---

*On page 109.

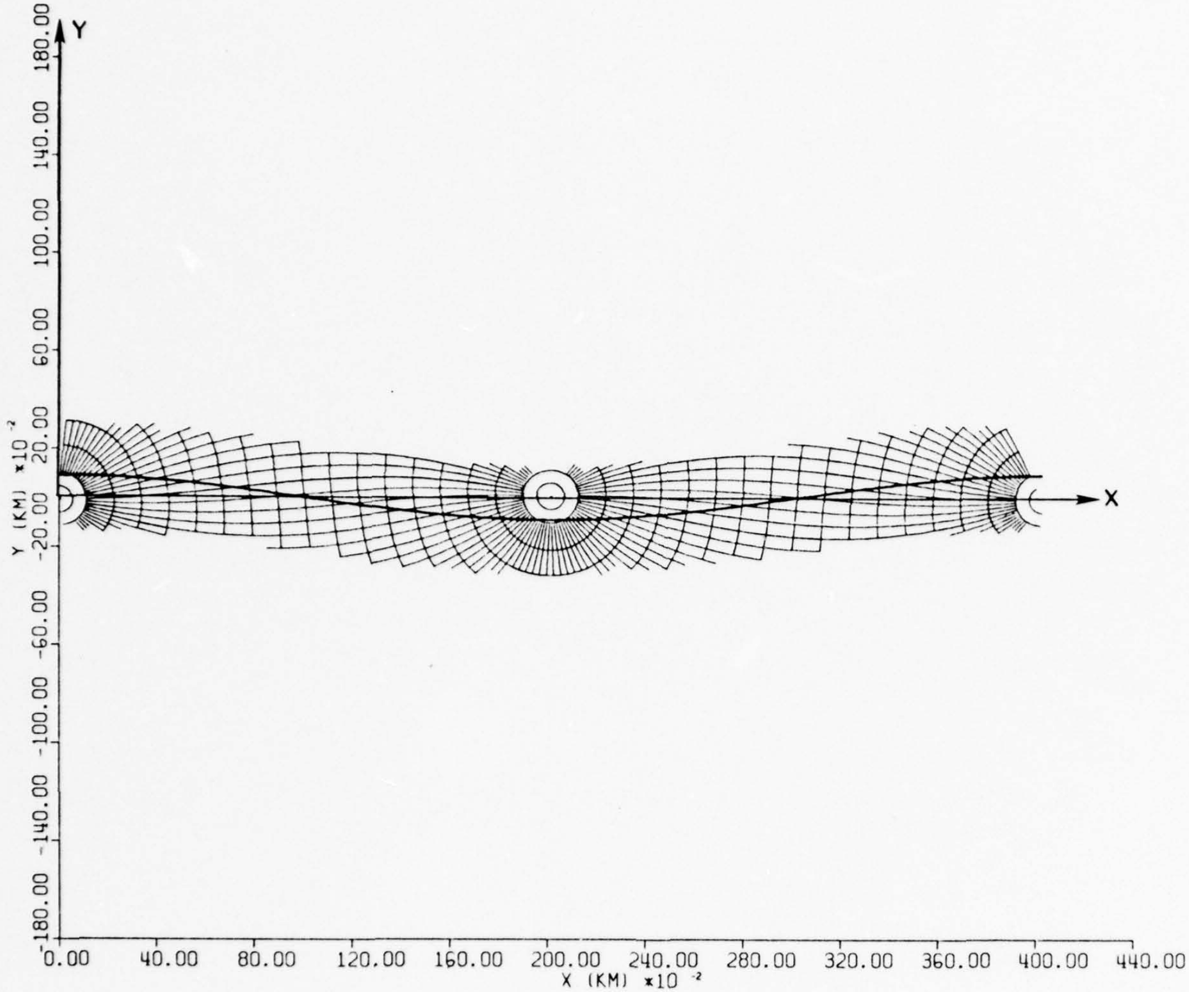


Figure 5.2 The LANDSAT SOM Graticule

When substituted into the inverse transformation (as discussed in §4.2 and implemented in Appendix D), the above (x,y) values result in recovery of the original (ϕ,λ) values (to within 10^{-11}). The printouts of the "Forward and Inverse Test Cases" of Appendix D supply values of intermediate quantities en-route to these end results.

5.3 The LANDSAT Graticule

By simply selecting a sequence of closely spaced (ϕ,λ) values along lines of constant ϕ and λ, the forward transformation equations of §4.1 (as implemented in Appendix D) result in the projections of the meridians and parallels into the map plane. The resulting graticule is displayed in Figure 5.2. Except for $\phi = \pm 85^\circ$, the parallels are plotted at a 10° interval (-80°, -70°, -60°, 70°, 80°) and the meridians are plotted at a 5° interval. The region displayed is for roughly $\pm 20^\circ$ (central angle) of the groundtrack.

Observe, qualitatively, that angles are well-preserved, even 20° off the groundtrack (meridians and parallels intersect at right angles). However, observe that slight shape distortions are evident in the departure of the 80° and 85° parallels from circles. As is evident in the formulation herein (and as is clear in the numerical error analyses of §5.4), rigorous satisfaction of constant scale and conformality are achieved along the groundtrack. These distortions off the groundtrack are entirely satisfactory within $\pm$ 100 km of the groundtrack (the approximate length of the LANDSAT scan-lines).

5.4 Length Distortion Analysis

Adopting the notation

s = arc length measured along some line on the ellipsoid

s' = arc length measured along the corresponding line in the map plane

then the basic equations for length distortion analysis are

$$\left(\frac{\partial s'}{\partial s}\right)_\lambda = \frac{(1-e^2 \sin^2\phi)^{3/2}}{a(1-e^2)} \left[\left(\frac{\partial x}{\partial \phi}\right)^2 + \left(\frac{\partial y}{\partial \phi}\right)^2\right]^{\frac{1}{2}} \tag{5.3a}$$

= local length distortion along a meridian (line of λ=constant)

$$\left(\frac{\partial s'}{\partial s}\right)_\phi = \frac{(1-e^2 \sin^2\phi)^{\frac{1}{2}}}{a} \left[\left(\frac{\partial x}{\partial \lambda}\right)^2 + \left(\frac{\partial y}{\partial \lambda}\right)^2\right]^{\frac{1}{2}} \tag{5.3b}$$

= local length distortion along a parallel (line of ϕ=constant).

The partial derivatives

$$\begin{bmatrix} \frac{\partial x}{\partial \phi} & \frac{\partial x}{\partial \lambda} \\ \frac{\partial y}{\partial \phi} & \frac{\partial y}{\partial \lambda} \end{bmatrix}$$

are developed in §6.4.

Evaluation of equations (5.3) at various points along and near the groundtrack resulted in the table of test case 4 (Appendix D). As is evident, the absolute length distortions are zero along the groundtrack, worst case errors of less than 3 parts per 10,000 occur at the outer fringe of the sensed region ($\pm$ 1°).

6.0 DETAILED FORMULATION OF THE MAP PROJECTION

6.1 Preliminary Remarks

In this section, we present the derivation of the map projection roughly in the manner it was developed originally, (sans the unproductive blind alleys!) attempting to expose major features of the logical process underlying the derivation. This logical process (and the resulting mathematics) partitions naturally into two major steps.

With reference to Figure 6.1, the first step is to project the satellite sub-point path (groundtrack) from the reference ellipsoid onto the mapping plane. This transformation should be such that the groundtrack length is not subject to local length distortions (zero scale distortion) and the "shape" of the groundtrack should be preserved (curvature constraint) at every point. These two objectives lead immediately to two corresponding differential equations which are developed and solved in §6.2.

With reference to Figure 6.2, the second step is the projection of all sensed points in a finite region on the ref. ellipsoid, centered on the groundtrack (i.e., the "sensed ribbon" on the earth's surface) into the mapping plane. This problem is approached using the intuitively clear notion that small displacements are made on the ellipsoid from a "locally nearly straight line" (the groundtrack; the radius of curvature of the groundtrack's tangent plane projection for LANDSAT orbits varies from over three earth radii to infinity, for example). This logic led us to consider the idealization that displacements near the groundtrack to nearby points might be well-approximated by displacements near the

Project Sub-Satellite Point Path G

From Ellipsoid to Map Plane

- Zero Length Distortion
- "Shape" Preserving

$t \rightarrow (\phi_g, \lambda_g) \rightarrow (x_g, y_g)$

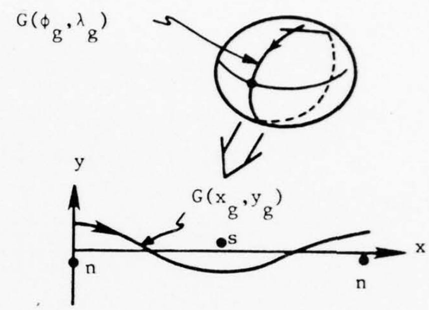


Figure 6.1 Projection of Sub-point from the Ellipsoid to the Map Plane

Project (ϕ, λ) Near the Sub-Satellite Path G

From the Ellipsoid to Corresponding (x,y) in the Map Plane

- Zero Length Distortion at G
- Rigorous Conformality at G
- Small Length Distortions within 10° of G
- Small Angle Distortions within 10° of G

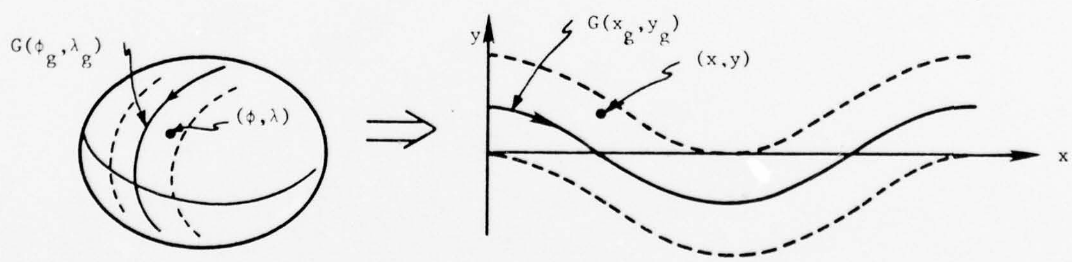


Figure 6.2 Projection of an Arbitrary Point (ϕ, λ) on the Ellipsoid to the Corresponding Point (x,y) in the Map Plane

equator of an oblique mercator projection (where the oblique equator is locally tangent to the groundtrack). The resulting map projection (based upon this idealization) is developed such that rigorous conformality and length preservation are satisfied only along the groundtrack, but the approximation is excellent within several hundred km of typical satellite groundtracks. Since this intuitively appealing approximation has worked out so well, we have not pursued the possibility that a more rigorous approximation to conformality and length preservation may be feasible (we did devote sufficient attention to this question, however to rule out the existence of exact conformal solutions for any cases other than trivial cases such as equatorial orbits or inclined orbits about a non-rotating spherical earth!). We develop the geometry and equations for the local mercator approximations in §6.3. These, in conjunction with the groundtrack projection equations of §6.2, are the essential formula of the desired map projection.

In §6.4, we develop the equations necessary to rigorously compute the partial derivatives

$$\Phi = \begin{bmatrix} \frac{\partial x}{\partial \phi} & \frac{\partial x}{\partial \lambda} \\ \\ \frac{\partial y}{\partial \phi} & \frac{\partial y}{\partial \lambda} \end{bmatrix} \tag{6.1}$$

which are necessary to do distortion analysis (§5.2) and which are required in the inverse transformation (§4.2) from given (x,y) in the map plane to the corresponding (ϕ,λ) on the ellipsoid.

6.2 Length and Shape Preserving Projection of the Satellite Groundtrack

First we state two desired constraints which lead directly to the projection of the sub-point path G from the ellipsoid to the map plane

(see Fig. 6.1). Observe that a transformation is desired from (ϕ_g, λ_g, t) along a line on the ellipsoid to the corresponding (x_g, y_g, t) along a line in the map plane. The line is generated as t varies from zero to an orbital period. Since a unique "two-to-two" mapping is desired, it is clear that two independent constraints are necessary and sufficient to establish the desired transformation.

Constant Scale Constraint

Let

s = arc length measured along the satellite sub-point trace (groundtrack)

$$= \int_0^t V(t)\, dt \tag{6.2}$$

s' = arc length measured along the projection of the groundtrack onto the map plane

$$= \int_0^t \sqrt{\left(\frac{dx_g}{d\tau}\right)^2 + \left(\frac{dy_g}{d\tau}\right)^2}\, d\tau. \tag{6.3}$$

We require, for no scale distortion along the groundtrack, that s=s' for all values of t. By inspection of equations (6.2) and (6.3), this is possible only if

$$V^2 = \left(\frac{dx_g}{dt}\right)^2 + \left(\frac{dy_g}{dt}\right)^2 \tag{6.4}$$

is satisfied at every point (x_g, y_g, t) along the map plane projection of the groundtrack.

Curvature Constraint

To preserve the groundtrack "shape", we require that the radius of curvature of the groundtrack projection (in the map plane) equal at

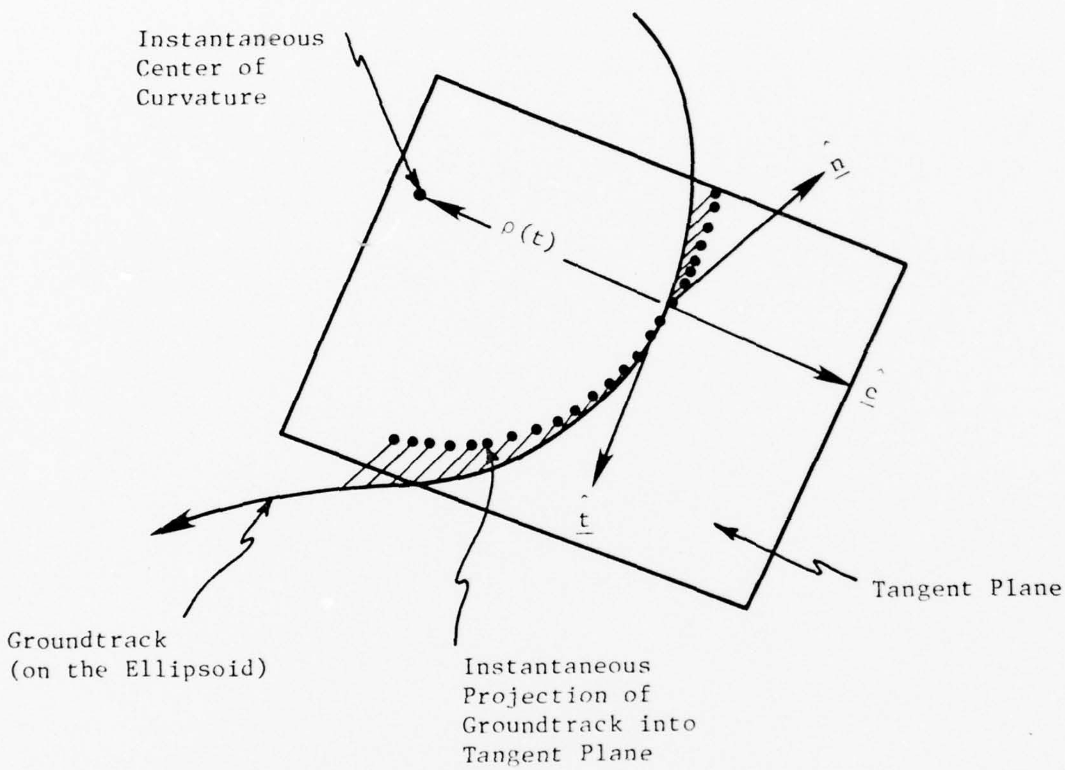


Figure 6.3 Radius of Curvature in the Tangent Plane

every point the instantaneous radius of curvature (ρ) of the projection of the groundtrack into the plane tangent to the ellipsoid at the sub-point (see Figure 6.3). This requires that

$$\left(\frac{d^2x_g}{ds^2}\right)^2 + \left(\frac{d^2y_g}{ds^2}\right)^2 = \frac{1}{\rho^2} \qquad (6.5)$$

Solution of eqns. (6.4) and (6.5) for the Groundtrack Projection. To change the independent variable of eqn.(6.4) from t to s, observe from eqn. (6.2) that

$$\frac{ds}{dt} = V \qquad (6.6a)$$

$$\frac{dx_g}{dt} = \frac{dx_g}{ds}\frac{ds}{dt} = V\frac{dx_g}{ds} \qquad (6.6b)$$

$$\frac{dy_g}{dt} = \frac{dy_g}{ds}\frac{ds}{dt} = V\frac{dy_g}{ds} \qquad (6.6c)$$

so that the length constraint can be written as simply

$$\left(\frac{dx_g}{ds}\right)^2 + \left(\frac{dy_g}{ds}\right)^2 = 1 \qquad (6.6d)$$

Thus we require a simultaneous solution for x_g and y_g satisfying equations (6.5) and (6.6d). By inspection of (6.6d) it is clear that the solution for the derivatives can be taken as simply

$$\frac{dx_g}{ds} = \cos f \qquad (6.7a)^*$$

$$\frac{dy_g}{ds} = \sin f \qquad (6.7b)^*$$

---

*Strictly speaking $\pm$ signs should accompany these equations; numerical experiments support the conclusion that the positive signs adopted are correct; note that the sign of the right hand side of eqn. (6.7) can change with the variation of ρ (see Appendix B).

where f is an appropriate function to guarantee satisfaction of the curvature constraints eqn (6.5). Substitution of (6.7) into (6.5) leads immediately to the conclusion that the first derivative of f must satisfy

$$\frac{df}{ds} = \frac{1}{\rho} \qquad (6.7c)^*$$

Making use of eqn. (6.6a) eqns. (6.7) can be written as the differential equations

$$\dot{x}_g \equiv \frac{dx_g}{dt} = V\cos f \qquad (6.8a)$$

$$\dot{y}_g \equiv \frac{dy_g}{dt} = V\sin f \qquad (6.8b)$$

$$\dot{f} \equiv \frac{df}{dt} = \frac{V}{\rho} \qquad (6.8c)$$

The solution of which is indicated formally as

$$f(t) = \int_o^t \frac{V(\tau)}{\rho(\tau)} \, d\tau \qquad (6.9a)$$

$$x_g(t) = 0 + \int_o^t V(\tau) \cos f(\tau) \, d\tau \qquad (6.9b)$$

$$y_g(t) = y_g(o) + \int_o^t V(\tau) \sin f(\tau) \, d\tau \qquad (6.9c)$$

where $V(t)$ and $\rho(t)$ are calculated as established in Appendices A and B.

It is obvious that the scale and curvature constraints, in the form of equations (6.4) and (6.5) are satisfied by equation (6.8) and therefore equations (6.9)

As is discussed in §4.4.1, the evaluation of the integrals in

eqn. (6.9) (if required for many t values) is greatly facilitated by replacing equations (6.9) by their fourier series. The integration of (6.9) can be done *sequentially* or *simultaneously*. Since the right hand sides of (6.9b) and (6.9c) contain f(t), f(t) can be integrated first from (6.9a) and then the (6.9b) and (6.9c) integrations can be carried out. Alternatively, we have found it much more convenient to do the integrations by using the Runge-Kutta algorithm (Appendix C) to integrate all three of equations (6.8) *simultaneously*; this is the method recommended. [If a large number of (ϕ,λ) points need to be transformed to the corresponding (x,y), the calculation of the Fourier series, discussed in §4.1.1, is strongly recommended]. The software of Appendix D implements the Runge-Kutta integration procedure to determine the coefficients of the Fourier series expansions for f(t), $x_g(t)$, and $y_g(t)$.

6.3 Local Oblique Mercator Projection of the Sensed Region from the Ellipsoid to the Map Plane

Since the scanned (sensed) points represent small displacements off the satellite groundtrack (which is typically relatively straight on a local scale, the LANDSAT orbit's radius of curvature component in the tangent plane varies from several earth radii to infinity), one is motivated to consider approximations to account for the small displacements from the *rigorously* projected groundtrack. We will now discuss local approximations which, together with §6.2 complete the map projection.

6.3.1 Scan Time Determination

Peculiar to the *dynamic* map projection under consideration is the

necessity to associate a particular time with the projection of (ϕ,λ) into the corresponding (x,y). In particular, this time is denoted t^*, it is the instant that the satellite scan vector passed over the earth-fixed point (ϕ,λ), (see Fig. 6.4d). Assuming infinite scan rate, this instant is characterized by the fact that the point (ϕ,λ) and the satellite sub-point (ϕ_g,λ_g) must lie in the plane determined by the vector $\hat{\underline{n}}$ (normal to the ellipsoid) and the vector $\hat{\underline{w}}$ (the scan vector, nomially normal to the orbit as seen in inertial space). This condition leads immediately to the constraint:

$$F(t^*) = \left[(\hat{\underline{w}} \times \hat{\underline{n}})\cdot\Delta\underline{R}\right]_{t=t^*} = 0 \tag{6.10}$$

where the time varying vectors

$$\hat{\underline{n}} = n_x\,\hat{\underline{i}} + n_y\,\hat{\underline{j}} + n_z\,\hat{\underline{k}} \tag{6.11}$$

$$\hat{\underline{w}} = w_x\,\hat{\underline{i}} + w_y\,\hat{\underline{j}} + w_z\,\hat{\underline{k}}$$

are computed according to formulas given in Appendices A and B and

$$\Delta\underline{R} = (R_x - R_{xg})\,\hat{\underline{i}} + (R_y - R_{yg})\,\hat{\underline{j}} + (R_z - R_{zg})\,\hat{\underline{k}} \tag{6.13}$$

$= $ the vector from (ϕ_g,λ_g) to (ϕ,λ)

and

$$R_{xg} = x - H_x,\ R_{yg} = y - H_y,\ R_{zg} = z - H_z$$

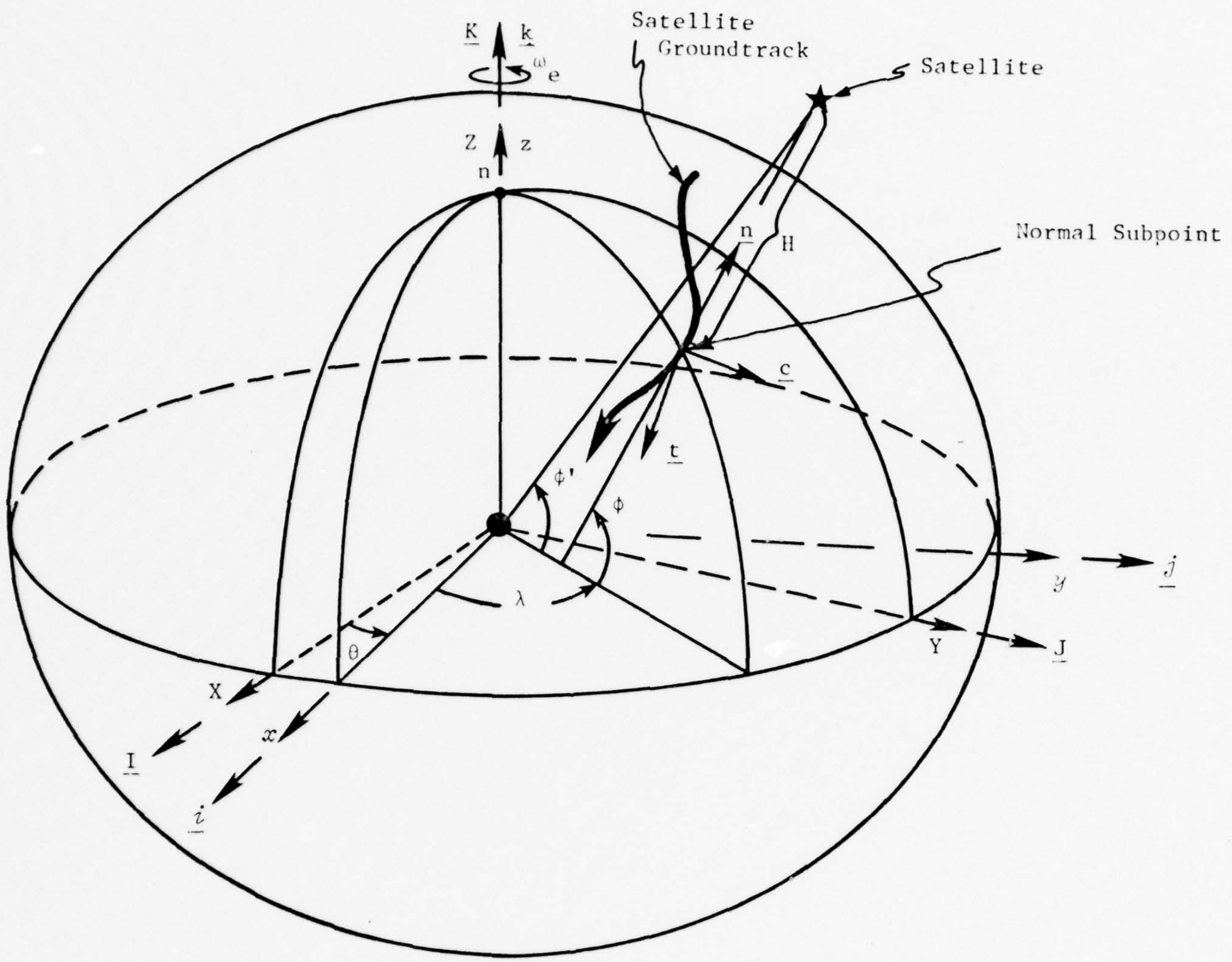
(x,y,z) earth-fixed components of the satellite position vectors, calculated from eqn. (A16).

(H_x,H_y,H_z) earth-fixed components of the satellite height vector, calculated from eqns. (B.2).

$$R_x = a^2(a^2\cos^2\phi + b^2\sin^2\phi)^{-\frac{1}{2}}\cos\phi\ \cos\lambda$$

$$R_y = a^2(a^2\cos\phi + b^2\sin^2\phi)^{-\frac{1}{2}}\cos\phi\ \sin\lambda \tag{6.14}$$

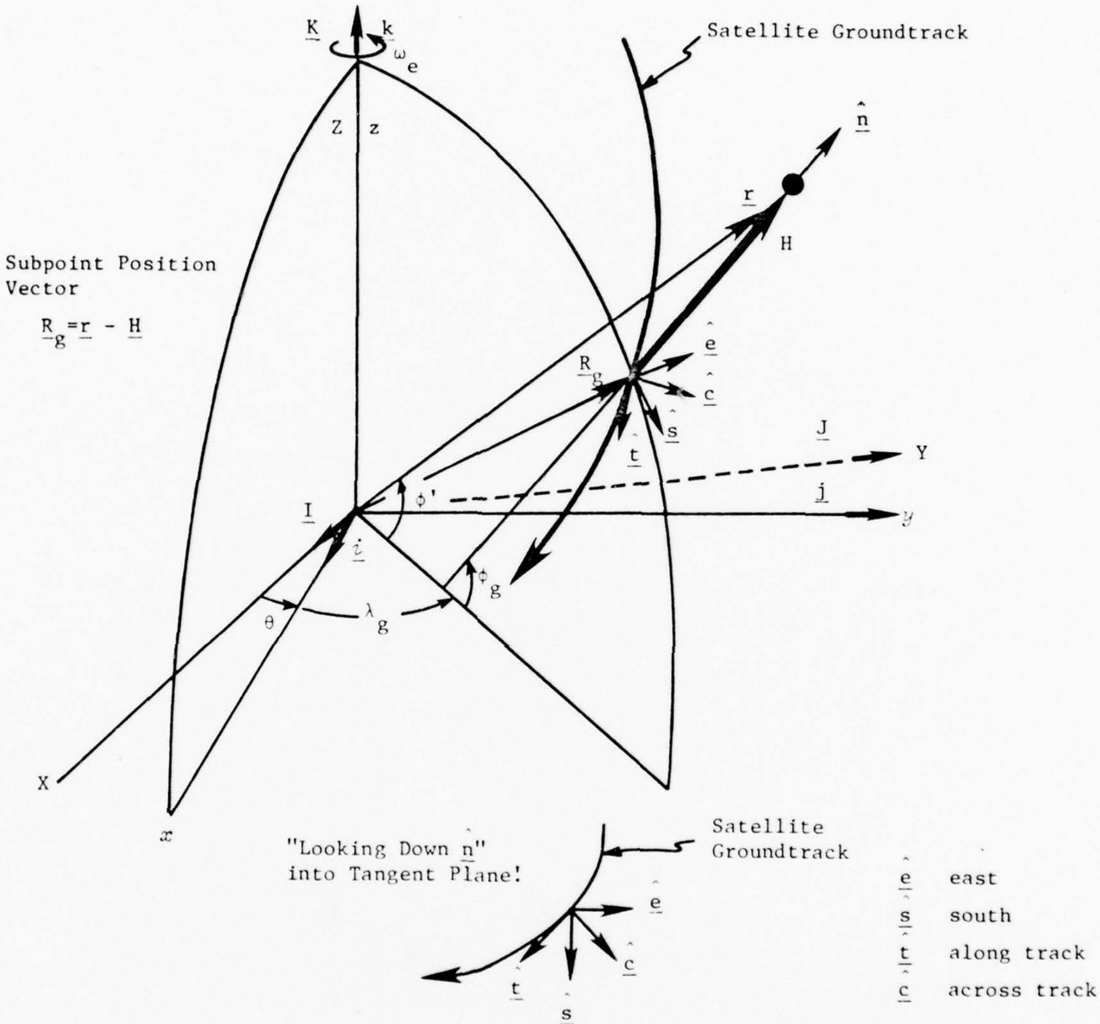
$$R_z = b^2(a^2\cos^2\phi + b^2\sin^2\phi)^{-\frac{1}{2}}\sin\phi$$



{X,Y,Z} Space Fixed Axes

{x,y,z} Earth Fixed Axes

Figure 6.4a Ellipsoidal Geometry

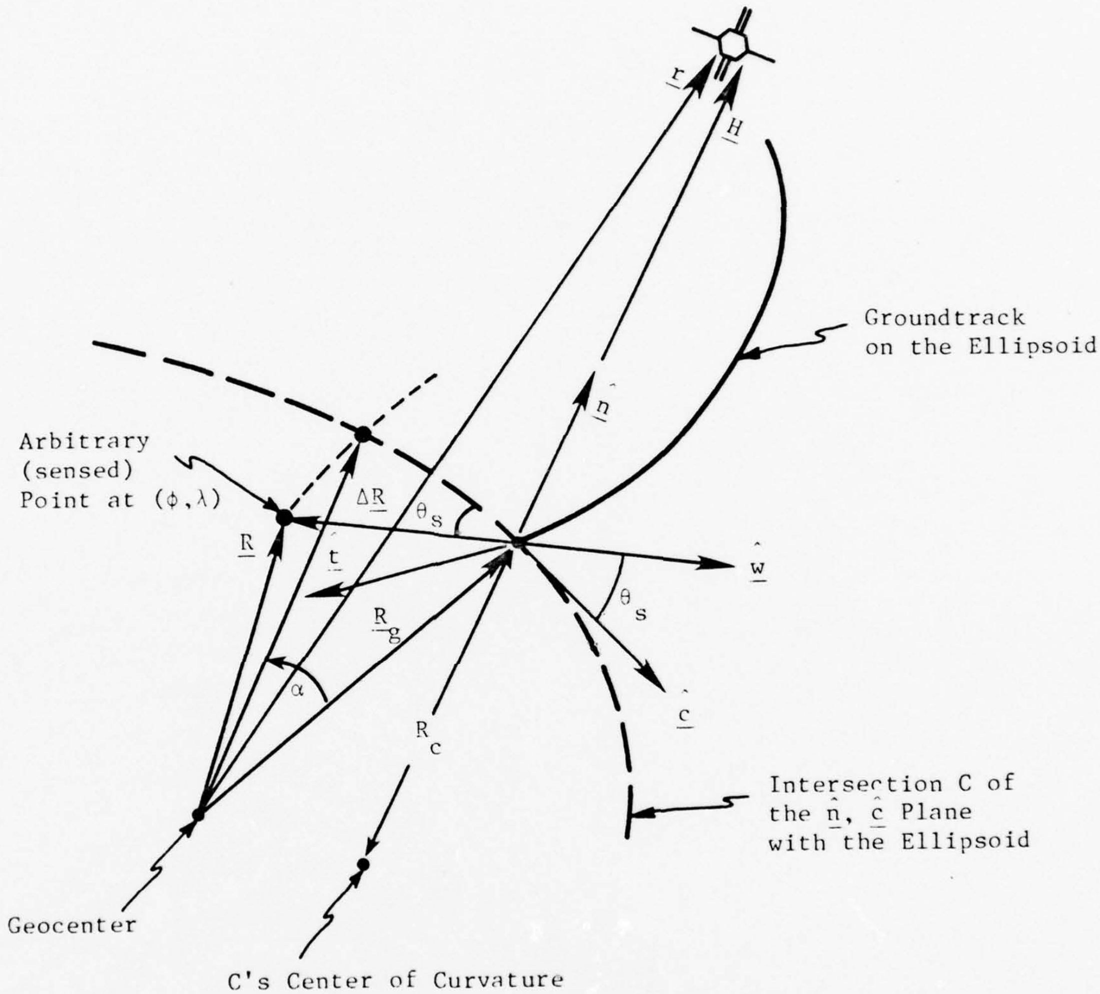


$$\hat{\underline{t}} = \frac{\dot{\underline{e}}_{R_g}}{\left|\dot{\underline{e}}_{R_g}\right|} = \frac{1}{V}\,\dot{\underline{e}}_{R_g}$$

$$\hat{\underline{c}} = \hat{\underline{n}} \times \hat{\underline{t}}$$

$$\begin{Bmatrix} \hat{\underline{s}} \\ \hat{\underline{e}} \\ \hat{\underline{n}} \end{Bmatrix} = \begin{bmatrix} \sin\phi_g \cos\lambda_g & \sin\phi_g \sin\lambda_g & -\cos\phi_g \\ -\sin\lambda_g & \cos\lambda_g & 0 \\ \cos\phi_g \cos\lambda_g & \cos\phi_g \sin\lambda_g & \sin\phi_g \end{bmatrix} \begin{Bmatrix} \underline{i} \\ \underline{j} \\ \underline{k} \end{Bmatrix}$$

Figure 6.4 B Ellipsoid Geometry: Tangent Plane Triads $\{\hat{\underline{s}}, \hat{\underline{e}}, \hat{\underline{n}}\}$ and $\{\hat{\underline{t}}, \hat{\underline{c}}, \hat{\underline{n}}\}$



$$R_c = \text{C's Radius of Curvature}$$
$$= (x - H_x)c\phi_g c\lambda_g + (y - H_y)c\phi_g s\lambda_g + (\frac{a}{b})^2 (z - H_z)s\phi_g$$
$$\alpha = \tan^{-1} (\frac{\hat{\underline{c}} \cdot \Delta\hat{\underline{R}}}{R_c})$$

Figure 6.4C Ellipsoidal Geometry: Ellipsoid Curvature in $\hat{\underline{n}}$, $\hat{\underline{c}}$ Plane

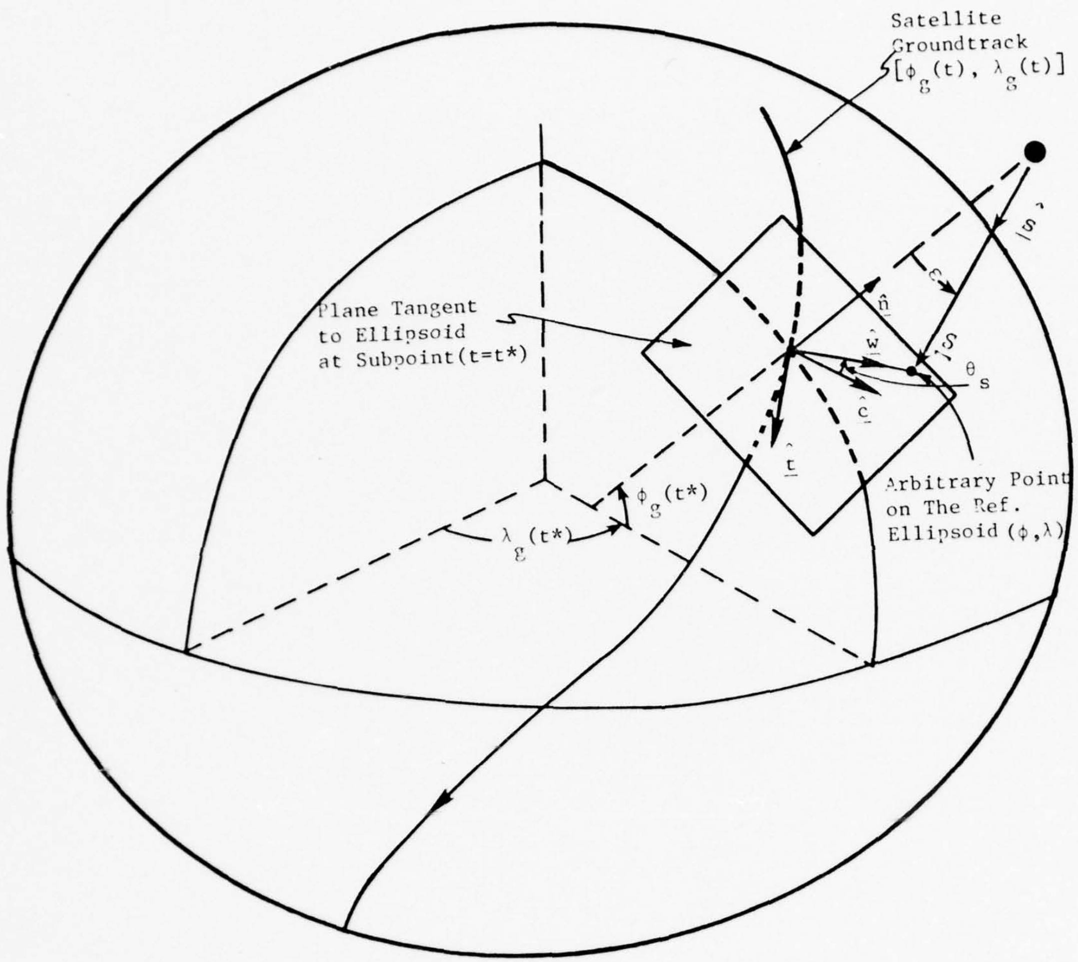


Figure 6.4 D Ellipsoidal Geometry: t^* = The Instant That the Scan Vector $\hat{\underline{w}}$ (Nominally, Normal to the Orbit Plane in Inertial Space) Passes Over the Given Point (ϕ,λ)

When earth-fixed components of the three vectors are substituted into eqn. (6.10), the function we seek to find the zero of is

$$F(t^*) = [(R_x + H_x - x)(w_y n_z - w_z n_y)$$

$$+ (R_y + H_y - y)(w_z n_x - w_x n_z)$$

$$+ (R_z + H_z - z)(w_x n_y - w_y n_x)]_{t=t^*} = 0 \qquad (6.15a)$$

and the time derivative $\dot{F}(t)$ follows immediately upon taking the derivative of (6.15a). Newton's method is used, as is discussed in §4.1.1 to find t* such that (6.15a) is satisfied. Usually convergence is achieved within 5 iterations.

Implicit in eqn. (6.15a) is the assumption that the scan is instantaneous (all points under the scan vector are imaged simultaneously). A rigorous calculation of the time varying location of the imaged point (ϕ,λ) requires a rigorous model for the scanner motion and the attitude motion. For our purposes here, it is sufficient to correct linearly for finite scan time by adding the correction

$$\Delta t = T\left(\frac{\varepsilon}{\varepsilon_{max}}\right) \qquad (6.15b)$$

to the root t* satisfying eqn. (6.15a), where

T = ½ the scan period

= 18.355 ms for the LANDSAT scanner

ε_{max} = the maximum scanner beam deflection away from local vertical

= 5.78 deg for the LANDSAT scanner

ε = instantaneous scanner deflection angle away from local vertical (see Fig. 6.4D)

$$= \cos^{-1}\left(\frac{-\underline{S}\cdot\underline{H}}{SH}\right) \cdot \text{sign}\,(\Delta\underline{R}\cdot\hat{\underline{c}}) \qquad (6.15c)$$

where $\underline{S} = \underline{R} - \underline{r}$

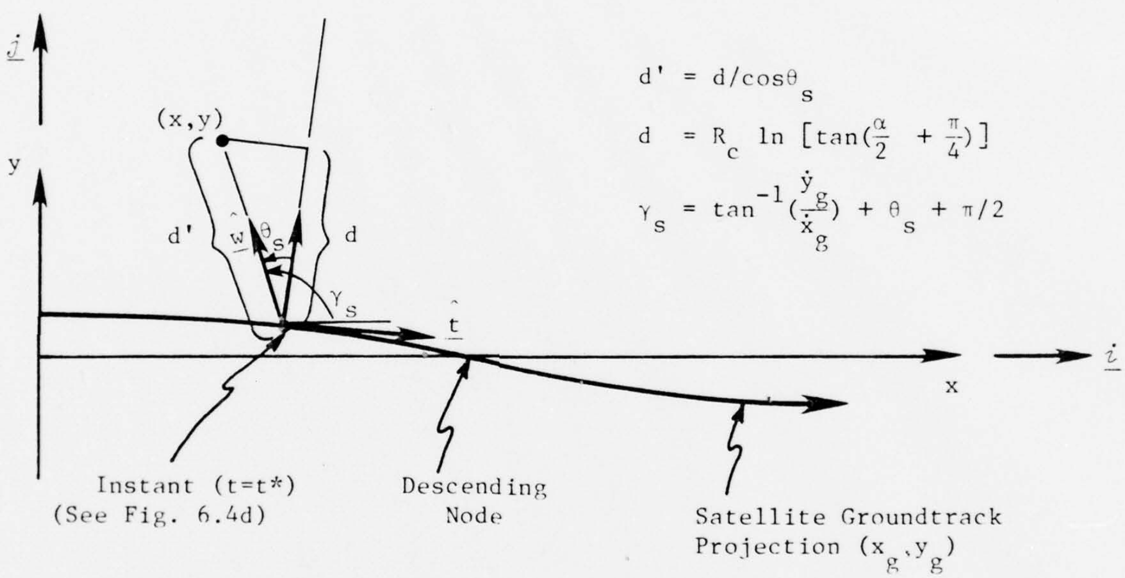


Figure 6.5 Map Plane Geometry: Local Mercator Approximation

6.3.2 Local Oblique Mercator Approximation

With particular reference to Figures 6.4d and 6.5, the equations relating a given (ϕ,λ) on the ellipsoid to the corresponding (x,y) in the map plane are

$$x = \int_o^{t*} V\cos f\, dt + R_c \ln[\tan(\frac{\pi}{4} + \frac{\alpha}{2})] \frac{\cos\gamma_s}{\cos\theta_s}$$

$$y = y_g(o) + \int_o^{t*} V\sin f\, dt + R_c \ln[\tan(\frac{\pi}{4} + \frac{\alpha}{2})] \frac{\sin\gamma_s}{\cos\theta_s} \tag{6.16}$$

where the integral terms are the groundtrack coordinates (x_g, y_g), developed in §6.2 and from the geometry of figures 6.4d and 6.5

$$\theta_s = - \sin^{-1}(\underline{\hat{w}}\cdot\underline{\hat{t}}) = - \sin^{-1}(w_x t_x + w_z t_z) = \text{scan angle} \tag{6.17a}$$

$$\gamma_s = \frac{\pi}{2} + \theta_s + \tan^{-1} (\frac{\dot{y}_g}{\dot{x}_g}) \tag{6.17b}$$

where

$\underline{\hat{w}}$ = scan vector, projection into $\underline{\hat{t}}$, $\underline{\hat{c}}$ plane or

$$\underline{w} = (\frac{\underline{w}'\cdot\underline{\hat{t}}}{u})\, \underline{\hat{t}} + (\frac{\underline{w}'\cdot\underline{\hat{c}}}{u})\underline{\hat{c}} \tag{6.18a}$$

$$u^2 = (\underline{\hat{w}}'\cdot\underline{\hat{t}})^2 + (\underline{\hat{w}}'\cdot\underline{\hat{c}})^2 \tag{6.18b}$$

$\underline{w}'$ = orbit normal, in inertial frame

$= \underline{r} \times {}^{n}\underline{\dot{r}}$ = constant, for unperturbed orbit(r = spacecraft position vector)* (6.18c)

and

R_c = ellipsoid radius of curvature in the $\underline{\hat{n}}$, $\underline{\hat{c}}$ plane

$$= (x - H_x)\cos\phi_g\cos\lambda_g + (y - H_y)\cos\phi_g\sin\lambda_g + \frac{a^2}{b^2} (z - H_z)\sin\phi_g \tag{6.19}$$

---

*See Appendix A

$$\alpha = \tan^{-1}\left(\frac{\hat{\underline{c}} \cdot \Delta\underline{R}}{R_c}\right) = \tan^{-1}\left(\frac{c_x(R_x - R_{xg}) + c_y(R_y - R_{yg}) + c_z(R_z - R_{zg})}{R_c}\right)$$

= the angle between $\hat{\underline{n}}$ and the vector from the cross-track center of curvature to the orthogonal projection of point (ϕ,λ) onto $\hat{\underline{c}}$ (see Figure 6.4c).

(c_x, c_y, c_z) = earth-fixed components of the cross track vector $\hat{\underline{c}}$, calculated from eqn. (B.8) of Appendix B.

Notice that the distance $R_c \ \ln[\tan(\frac{\pi}{4} + \frac{\alpha}{2})]$, measured normal to the groundtrack (Figure 6.5 and Eqns. 6.16), is identical to the classical formula for "y" of the transverse mercator projection (see References 2 and 3). Thus, to the extent that the groundtrack approximates a great circle, the map projection approximates an oblique mercator projection. However, the degree to which the classical oblique mercator map projection is approximated is not terribly important, in as much as the present projection is motivated by the fact that none of the classical projections fulfill the objectives being pursued here (primarily, a continuous distortion-free mapping of the entire sensed region of typical earth-scanning satellites). As is noted in §5.3, the length and angle distortions are sufficiently small to compare favorably with the

corresponding errors for the classical transverse mercator projection.

6.4 SOM Partial Derivatives (for Distortion Analysis and Inverse Transformations)

We summarize here the equations in back-substitution form for computation of the partial derivatives

$$\begin{bmatrix} \frac{\partial x}{\partial \phi} & \frac{\partial x}{\partial \lambda} \\ \frac{\partial y}{\partial \phi} & \frac{\partial y}{\partial \lambda} \end{bmatrix} \tag{6.20}$$

which are necessary in the calculation of §4.2, 5.2 and 5.4.

To compact the notation, we will make use of the symbol "$\phi \to \lambda$" whenever the identical equation for λ results by simply replacing ϕ by λ. These equations follow by simply applying chain partial differentiation to the equations of §6.2 and 6.3. While references are frequently made to the equations differentiated to obtain these results, the present treatment is more nearly an "annotated summary" rather than a rigorous derivation.

Taking partial derivatives of equations (6.16) yield

$$\begin{aligned} \frac{\partial x}{\partial \phi} &= \frac{\partial x_g}{\partial \phi} + \frac{\partial D}{\partial \phi} \cos\gamma_s - \frac{\partial \gamma_s}{\partial \phi} D \sin\gamma_s, \qquad \phi \to \lambda \\ \frac{\partial y}{\partial \phi} &= \frac{\partial y_g}{\partial \phi} + \frac{\partial D}{\partial \phi} \sin\gamma_s + \frac{\partial \gamma_s}{\partial \phi} D \cos\gamma_s, \qquad \phi \to \lambda \end{aligned} \tag{6.21}$$

where

$$\begin{aligned} x_g &\equiv \int_o^{t^*} V\cos f\, dt \\ y_g &\equiv y_g(o) + \int_o^{t^*} V\sin f\, dt \end{aligned} \tag{6.22}$$

$$D \equiv R_c \ln[\tan(\tfrac{\alpha}{2} + \tfrac{\pi}{4})]/\cos\theta_s \tag{6.23}$$

The partials $\left(\frac{\partial x_g}{\partial\phi}, \frac{\partial y_g}{\partial\phi}, \frac{\partial D}{\partial\phi}, \frac{\partial\gamma_s}{\partial\phi}; \phi\to\lambda\right)$ needed to calculate eqns. (6.21) are determined as follows:

<u>x_g, y_g partials</u>

from eqns. (6.22)

$$\frac{\partial x_g}{\partial\phi} = \frac{\partial x_g}{\partial t^*}\,\frac{\partial t^*}{\partial\phi} = V(t^*)\cos[f(t^*)]\,\frac{\partial t^*}{\partial\phi}\ ,\quad \phi\to\lambda$$

$$\frac{\partial y_g}{\partial\phi} = \frac{\partial y_g}{\partial t^*}\,\frac{\partial t^*}{\partial\phi} = V(t^*)\sin[f(t^*)]\,\frac{\partial t^*}{\partial\phi}\ ,\quad \phi\to\lambda \tag{6.24}$$

<u>D Partials</u>

$$\frac{\partial D}{\partial\phi} = \frac{1}{\cos\theta_s}\left\{\frac{\partial R_c}{\partial\phi}\ln[\tan(\frac{\alpha}{2}+\frac{\pi}{4})] + \frac{\partial\alpha}{\partial\phi}\ \frac{R_c\ \sec^2(\frac{\pi}{4}+\frac{\alpha}{2})}{2\ \tan(\frac{\pi}{4}+\frac{\alpha}{2})}\right\}$$

$$- R_c\ln[\tan(\frac{\pi}{4}+\frac{\alpha}{2})]\ \frac{\sin\theta_s}{(\cos\theta_s)^2}\,\frac{\partial\theta_s}{\partial\phi}\ ,\quad \phi\to\lambda \tag{6.25}$$

where

$$\frac{\partial R_c}{\partial\phi} = [\frac{\partial R_c}{\partial\phi_g}\,\frac{\partial\phi_g}{\partial t^*} + \frac{\partial R_c}{\partial\lambda_g}\,\frac{\partial\lambda_g}{\partial t^*} + \frac{\partial R_x}{\partial t^*}\,c\phi_g c\lambda_g$$

$$+ \frac{\partial R_y}{\partial t^*}\,c\phi_g\ s\lambda_g + \frac{a^2}{b^2}\,\frac{\partial R_z}{\partial t^*}\,s\phi_g]\ \frac{\partial t^*}{\partial\phi}\ ,\quad \phi\to\lambda \tag{6.26}$$

$$\frac{\partial R_c}{\partial\phi_g} = -\ R_x s\phi_g\ c\lambda_g - R_y\ s\phi_g\ s\lambda_g + \frac{a^2}{b^2}\,R_z c\phi_g \tag{6.26}$$

$$\frac{\partial R_c}{\partial\lambda_g} = -\ R_x\ c\phi_g\ s\lambda_g + R_y\ c\phi_g\ c\lambda_g \tag{6.28}$$

$$\frac{\partial\alpha}{\partial\phi} = \frac{1}{R_c^{\ 2} + (\Delta\underline{R}\cdot\hat{\underline{c}})^2}\,\{R_c[\frac{\partial\Delta\underline{R}}{\partial\phi}\cdot\hat{\underline{c}} + \Delta\underline{R}\cdot\frac{\partial\hat{\underline{c}}}{\partial\phi}] - (\Delta\underline{R}\cdot\hat{\underline{c}})\frac{\partial R_c}{\partial\phi}\},\quad \phi\to\lambda \tag{6.29}$$

$$\frac{\partial\Delta\underline{R}}{\partial\phi} = (T_1,\ T_2,\ T_3) \tag{6.30}$$

with

$$T_1 = \frac{\partial N}{\partial \phi} c\phi c\lambda - Ns\phi c\lambda \tag{6.31a}$$

$$T_2 = \frac{\partial N}{\partial \phi} c\phi c\lambda - N s\phi s\lambda \tag{6.31b}$$

$$T_3 = \frac{b^2}{a^2} (\frac{\partial N}{\partial \phi} s\phi + Nc\phi) \tag{6.31c}$$

then

$$N = a^2(a^2c^2\phi + b^2s^2\phi)^{-\frac{1}{2}} \tag{6.32}$$

$$\frac{\partial N}{\partial \phi} = - N^3 \frac{(a^2 - b^2)}{2a^4} \sin(2\phi) \tag{6.33}$$

$$\frac{\partial \Delta \underline{R}}{\partial \lambda} = (S_1, S_2, S_3) \tag{6.34}$$

with

$$S_1 = - N c\phi s\lambda \tag{6.35a}$$

$$S_2 = N c\phi c\lambda \tag{6.35b}$$

$$S_3 = 0$$

then

$$\frac{\partial \hat{\underline{c}}}{\partial \phi} = \frac{\partial}{\partial \phi} (\hat{\underline{n}} \times \hat{\underline{t}}) = \frac{\partial \hat{\underline{n}}}{\partial \phi} \times \hat{\underline{t}} + \hat{\underline{n}} \times [\frac{1}{V} \frac{{}^e d^2 \underline{R}}{dt^{*2}} - \frac{1}{V^2} \frac{{}^e d\underline{R}}{dt^*} \frac{dV}{dt^*}] \frac{\partial t^*}{\partial \phi} , \phi \to \lambda \tag{6.36}$$

$$\frac{\partial \hat{\underline{n}}}{\partial \phi} = [\frac{\partial \hat{\underline{n}}}{\partial \phi_g} \frac{\partial \phi_g}{\partial t^*} + \frac{\partial \hat{\underline{n}}}{\partial \lambda_g} \frac{\partial \lambda_g}{\partial t^*}] \frac{\partial t^*}{\partial \phi} , \phi \to \lambda \tag{6.37}$$

$$\frac{\partial \hat{\underline{n}}}{\partial \phi_g} = (T_1, T_2, T_3) \tag{6.38}$$

with

$$T_1 = -s\phi_g \, c\lambda_g \tag{6.39a}$$

$$T_2 = -s\phi_g \; s\lambda_g \qquad (6.39b)$$

$$T_3 = c\phi_g \qquad (6.39c)$$

then

$$\frac{\partial \hat{\underline{n}}}{\partial \lambda_g} = (S_1, S_2, S_3) \qquad (6.40)$$

with

$$S_1 = - c\phi_g \; s\lambda_g \qquad (6.41a)$$

$$S_2 = c\phi_g \; c\lambda_g \qquad (6.42b)$$

$$S_3 = 0 \qquad (6.43c)$$

then

$$\frac{\partial V}{\partial t^*} = \frac{1}{V} \left[\frac{{}^e d\underline{R}}{dt} \cdot \frac{{}^e d^2\underline{R}}{dt^2}\right]_{t=t^*} \qquad (6.44)$$

From eqn. (6.17a)

$$\frac{\partial \theta_s}{\partial \phi} = - \frac{1}{1 - (\hat{\underline{w}} \cdot \hat{\underline{t}})^{2\,\frac{1}{2}}} \left[\frac{\partial \hat{\underline{w}}}{\partial \phi} \cdot \hat{\underline{t}} + \hat{\underline{w}} \cdot \frac{\partial \hat{\underline{t}}}{\partial \phi}\right], \; \phi \rightarrow \lambda \qquad (6.45)$$

From Eqn. (6.18)

$$\frac{\partial \hat{\underline{w}}}{\partial \phi} = \frac{\partial \hat{\underline{w}}}{\partial t^*} \; \frac{\partial t^*}{\partial \phi} \quad , \; \phi \rightarrow \lambda \qquad (6.46)$$

See eqn. (6.62) for $\frac{\partial \hat{\underline{w}}}{\partial t^*}$

$$\frac{\partial \underline{t}}{\partial \phi} = \frac{\partial}{\partial \phi} \left[\frac{1}{V} \; \frac{{}^e d\underline{R}}{dt^*}\right] = \frac{1}{V} \left[\frac{{}^e d^2\underline{R}}{dt^{*2}} - \frac{1}{V} \; \frac{\partial V}{\partial t^*} \; \frac{{}^e d\underline{R}}{dt^*}\right] \frac{\partial t^*}{\partial \phi}, \; \phi \rightarrow \lambda \qquad (6.47)$$

γ_s Partials

From eqn. (6.17b)

$$\frac{\partial \gamma_s}{\partial \phi} = \frac{\partial \theta_s}{\partial \phi} + \frac{\partial}{\partial \phi} \left[\tan^{-1} \; \left(\frac{\dot{y}_g}{\dot{x}_g}\right)\right]$$

or

$$\frac{\partial \gamma_s}{\partial \phi} = \frac{\partial \theta_s}{\partial \phi} + \left[\left(\frac{dx_g}{dt}\right)^2 + \left(\frac{dt_g}{dt}\right)^2 \right]^{-1} \left\{ \frac{dx_g}{dt^*} \frac{d^2 y_g}{dt^{*2}} - \frac{dy_g}{dt^*} \frac{d^2 x_g}{dt^{*2}} \right\} \frac{\partial t^*}{\partial \phi} \ , \quad \phi \to \lambda \tag{6.48}$$

with

$$\frac{d^2 x_g}{dt^{*2}} = \frac{dV(t^*)}{dt^*} \cos\,(f(t^*)) - V(t^*) \sin\,(f(t^*)) \frac{df(t^*)}{dt^*} \tag{6.49a}$$

$$\frac{d^2 y_g}{dt^{*2}} = \frac{dV(t^*)}{dt^*} \sin\,(f(t^*)) + V(t^*) \cos\,(f(t^*)) \frac{df(t^*)}{dt^*} \tag{6.49b}$$

From the t* derivative of eqn. (6.10)

$$\frac{\partial t^*}{\partial \phi} = - \left(\frac{\partial F}{\partial t^*}\right)^{-1} \frac{\partial F}{\partial \phi} \ , \quad \phi \to \lambda \tag{6.50}$$

when

$$F(t^*) = [\underline{\hat{w}} \times \underline{\hat{n}}) \cdot \Delta \underline{R}]_{t=t^*} \ ; \ \underline{\hat{t}}_p \equiv \underline{\hat{w}} \times \underline{\hat{n}}$$

$$= \underline{\hat{t}}p \cdot \Delta \underline{R} \ ;, \quad \Delta \underline{R} \equiv (\Delta \underline{R} \cdot \underline{\hat{t}})\, \underline{\hat{t}} + (\Delta \underline{R} \,.\, \underline{\hat{c}})\, \underline{\hat{c}}$$

then

$$F(t^*) = (\underline{\hat{t}}_p \cdot \underline{\hat{t}})(\Delta \underline{R} \cdot \underline{\hat{t}}) + (\underline{\hat{t}} \cdot \underline{\hat{c}})(\Delta \underline{R} \cdot \underline{\hat{c}})$$

and

$$\frac{\partial F}{\partial \phi} = \left(\frac{\partial (\Delta \underline{R})}{\partial \phi} \cdot \underline{\hat{t}}\right)(\underline{\hat{t}} \cdot \underline{\hat{t}}p) + \left(\frac{\partial (\Delta \underline{R})}{\partial \phi} \cdot \underline{\hat{c}}\right)(\underline{\hat{c}} \cdot \underline{\hat{t}}p), \quad \phi \to \lambda \tag{6.51}$$

$$\frac{\partial F}{\partial t^*} = \left\{ \left(\frac{\partial (\Delta \underline{R})}{\partial t^*} \cdot \underline{\hat{t}}\right) + \left(\Delta \underline{R} \cdot \frac{\partial \underline{\hat{t}}}{\partial t^*}\right) \right\} (\underline{\hat{t}} \cdot \underline{\hat{t}}p) + (\Delta \underline{R} \cdot \underline{\hat{t}}) \left\{ \left(\frac{\partial \underline{\hat{t}}}{\partial t^*} \cdot \underline{\hat{t}}p\right) + \left(\underline{\hat{t}} \cdot \frac{\partial \underline{\hat{t}}p}{\partial t^*}\right) \right\}$$

$$+ \left\{ \left(\frac{\partial (\Delta \underline{R})}{\partial t^*} \cdot \underline{\hat{c}}\right) + \left(\Delta \underline{R} \cdot \frac{\partial \underline{\hat{c}}}{\partial t^*}\right) \right\} (\underline{\hat{c}} \cdot \underline{\hat{t}}p) + (\Delta \underline{R} \cdot \underline{\hat{c}}) \left\{ \left(\frac{\partial \underline{\hat{t}}}{\partial t^*} \cdot \underline{\hat{t}}p\right) + \left(\underline{\hat{c}} \cdot \frac{\partial \underline{\hat{t}}p}{\partial t^*}\right) \right\}$$

$$\frac{\partial \hat{\underline{t}}_p}{\partial t^*} = \frac{\partial \hat{\underline{n}}}{\partial t^*} \times \hat{\underline{w}} + \hat{\underline{n}} \times \frac{\partial \hat{\underline{w}}}{\partial t^*} \tag{6.53}$$

$$\hat{\underline{w}}' = \underline{r} \times {}^n\dot{\underline{r}} \qquad \text{(normal to orbit in the inertial frame)} \tag{6.54}$$

$$E\dot{\hat{\underline{w}}} = {}^n\dot{\hat{\underline{w}}}' + {}^n\underline{\omega}^e \times \hat{\underline{w}}' \;, \quad {}^n\underline{\omega}^e = \dot{\theta}\,\hat{\underline{k}} = \text{ang. vel. of earth} \tag{6.55}$$

$${}^n\dot{\hat{\underline{w}}} = \underline{r} \times {}^n\ddot{\underline{r}} \quad \text{(=0, for un-perturbed elliptical orbits)} \tag{6.56}$$

Given inertial components of $\dot{\hat{\underline{w}}}'$ and ${}^e\dot{\hat{w}}'$, the earth-fixed components are obtained via the transformation

$$\{\hat{\underline{w}}'\}_e = [E_3(\theta)]\{\hat{\underline{w}}'\}_n \tag{6.58}$$

$$\{{}^e\dot{\hat{w}}\}_e = [E_3(\theta)]\{{}^e\dot{\hat{w}}\}_n \tag{6.59}$$

$$[E_3(\theta)] = \begin{bmatrix} c\theta & s\theta & 0 \\ -s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.60}$$

$$\theta = \theta_o + \omega_e(t-t_o), \omega_e = \text{rotational rate of the earth} \tag{6.61}$$

then

$$\frac{\partial \hat{\underline{w}}}{\partial t^*} = A\,\hat{\underline{t}} + B\,\hat{\underline{c}} + c\,\frac{\partial \underline{c}}{\partial t^*} + D\,\frac{\partial \hat{\underline{t}}}{\partial t^*} \tag{6.62}$$

where

$$A = \frac{1}{u}\left[\dot{\hat{\underline{w}}}' \cdot \hat{\underline{t}} + \hat{\underline{w}}' \cdot \frac{\partial \hat{\underline{t}}}{\partial t}\right] - \frac{\hat{\underline{w}}' \cdot \hat{\underline{t}}}{u^3}\left\{\hat{\underline{w}}' \cdot \hat{\underline{t}}\left[\dot{\hat{\underline{w}}}' \cdot \hat{\underline{t}} + \hat{\underline{w}}' \cdot \frac{\partial \hat{\underline{t}}}{\partial t^*}\right] + \hat{\underline{w}}' \cdot \hat{\underline{c}}\left[\dot{\hat{\underline{w}}}' \cdot \underline{c} + \hat{\underline{w}}' \cdot \frac{\partial \underline{c}}{\partial t^*}\right]\right\}$$

$$B = \frac{1}{u}\left[\dot{\hat{\underline{w}}}' \cdot \hat{\underline{c}} + \hat{\underline{w}}' \cdot \frac{\partial \hat{\underline{c}}}{\partial t}\right] - \frac{\hat{\underline{w}}' \cdot \hat{\underline{c}}}{u^3}\left\{\hat{\underline{w}}' \cdot \hat{\underline{t}}\left[\dot{\hat{\underline{w}}}' \cdot \hat{\underline{t}} + \hat{\underline{w}}' \cdot \frac{\partial \hat{\underline{t}}}{\partial t^*}\right] + \hat{\underline{w}}' \cdot \hat{\underline{c}}\left[\dot{\hat{\underline{w}}}' \cdot \hat{\underline{c}} + \hat{\underline{w}}' \cdot \frac{\partial \hat{\underline{c}}}{\partial t^*}\right]\right\} \tag{6.64}$$

$$c = \frac{1}{u}\,\hat{\underline{w}}' \cdot \hat{\underline{c}} \tag{6.65}$$

$$D = \frac{1}{u}\,\hat{\underline{w}}' \cdot \hat{\underline{t}} \tag{6.66}$$

$$u^2 = (\hat{\underline{w}}' \cdot \hat{\underline{t}})^2 + (\hat{\underline{w}}' \cdot \hat{\underline{c}})^2 \tag{6.67}$$

7.0 CONCLUDING REMARKS

This report documents a fairly general formulation and implementation of the Space Oblique Mercator map projection which embody the attractive features forecast by Colvocoresses [(1)] when he first conceived of this projection. During the middle stages of this development, the authors became aware that a parallel effort was being carried out by John Snyder [(4)]. Based upon recent communications, it is clear that Snyder has accomplished an approximately equal feat, but using a different approach.*

Snyder's insightful formulation is much more compact than the present developments, although the one-time character of many of the calculations would appear to diminish the gap in computational efficiency.

The question of "which formulation should be used" may boil down to the issue of "how much flexibility is required vis-a-vis the nominal orbit". Should circular orbits be exclusively desired, then Snyder's formulation may prove preferable. Should a state-of-the art orbit integration program be used to define the nominal orbit, then, without question, the present formulation is applicable whereas Snyder's is not. Another important unresolved issue is how important

---

*Snyder has not rigorously enforced the curvature constraint (that the map plane projection of the groundtrack have the same radius of curvature as the tangent plane projection of the groundtrack), but his solution approximately satisfies this constraint nonetheless. Snyder's results do not allow as much generality with regard to selection of the nominal orbit. Whereas the present formulation permits routine use with either analytical or numerically integrated nominal orbits, Snyder's formulations were designed for circular orbits (although we understand that he is attempting to modify his formulation to account for non-circular orbits).

rigorous enforcement of the curvature constraint will prove for orbits other than the LANDSAT near-polar, near-circular, cases studied thus far.

In any event, we believe both Snyder's formulation and the present formulation are useful contributions; both will likely be employed in future utilizations of LANDSAT and similar imagry.

As a final remark, we note that the considerable logical processes, algebra, calculus, and computer programming underlying this work, coupled with the usual editorial/typographical headaches leave a finite probability that significant errors have escaped the attention of several proof-readers. We sincerely hope that any remaining errors prove of no conceptual or practical consequence, and we solicit the readers communication of all errors or suspected errors. It is anticipated that computational studies and further analysis will lead to addendums to this work in which all known errors will be corrected.

8.0 REFERENCES

1. Colvocoresses, A.P., "Space Oblique Mercator", Photogrammetric Engineering and Remote Sensing, V. 40, No. 8, pp. 921-26, 1974.

2. Thomas, P.D., Conformal Projections in Geodesy and Cartography, Spl. Publ. #251, U.S. Coast and Geodetic Survey, Washington, D.C. 1952.

3. Hotine, Brigadier M., "The Orthomorphic Projection of the Spheroid", The Empire Surevey Review, V. VIII and IX, No. 62-66, 1946-47.

4. Snyder, John P., private oral communications during summer of 1977, and written notes dated August 30, 1977, August 14, 1977 and October 3, 1977.

5. Herrick, Samuel, Astrodynamics, Vol. 1, Van Nostrand Reinhold, Ch. 1-4, 7, 1971.

6. Battin, R.H., Astronautical Guidance, McGraw-Hill, Ch. 1-4, 1964.

APPENDIX A

ORBIT INTEGRATION AND TRANSFORMATIONS

A.1 Comments

Here we consider two levels of generality for orbit calculation, viz:

(1) The case of arbitrary, but given, smooth perturbation, (non-2 body effects) and

(2) The Keplerian 2-body orbit (including all possible species of ellipital and circular orbits).

The first class of orbits clearly includes the latter as an obvious special case of zero perturbations. However, the well known analytical solution for the Keplerian case is so efficient (relative to numerically integrated orbits) that we treat it separately.

A.2 Equations of Motion, Inertial Coordinates

We observe first that the differential equations of motion for the general case have the form

$$\begin{aligned}\ddot{X} &= -GM\ X/r^3 + (X \quad \text{perturbation acceleration})\\ \ddot{Y} &= -GM\ Y/r^3 + (Y \quad \text{perturbation acceleration})\\ \ddot{Z} &= -GM\ Z/r^3 + (Z \quad \text{perturbation acceleration})\end{aligned} \tag{A.1}$$

where

$r = X\underline{I} + Y\underline{J} + Z\underline{K}$ = satellite rectangular coordinates in a non-rotating earth-centered frame, with Z along the polar axis.

$$r^2 = X^2 + Y^2 + Z^2 \tag{A.2}$$

GM = Earth's gravitation-mass constant = $398601.2\ \frac{km^3}{sec^2}$

Given initial position and velocity coordinates $(X_o, Y_o, Z_o, \dot{X}_o, \dot{Y}_o, \dot{Z}_o)$

at time t_o, numerical methods such as Runge-Kutta (Appendix C) can be employed to integrate equations (A.1) to determine instantaneous position and velocity $(X,Y,Z,\dot{X},\dot{Y},\dot{Z})$ at an arbitrary given time t.

A.3 Satellite Motion: Analytical Solution for Inertial Rectangular Coordinates (Keplerian Special Case)

The following equations, in order of solution, determine the inertial coordinates $(X,Y,Z,\dot{X},\dot{Y},\dot{Z})$ at time t, given the same quantities $(X_o, ---, \dot{Z}_o)$ at time t_o:

constants:

$\mu \equiv GM = 398601.2 \quad km^3/\ sec^2$

$$r_o^2 = X_o^2 + Y_o^2 + Z_o^2 \tag{A.3}$$

$$D_o = X_o\dot{X}_o + Y_o\dot{Y}_o + Z_o\dot{Z}_o \tag{A.4}$$

$$V_o^2 = \dot{X}_o^2 + \dot{Y}_o^2 + \dot{Z}_o^2 \tag{A.5}$$

$$\frac{1}{a} = 2/r_o - V_o^2/\mu \tag{A.6}$$

$$c_o = 1 - r_o/a \tag{A.7}$$

Solve for the change in eccentric anomoly $\hat{E}$ (using Newton's method) from

$$\sqrt{\mu}(t-t_o)a^{-3/2} = \hat{E} - (1-r_o/a)\ \sin\hat{E} + \frac{D_o}{\sqrt{\mu a}}\ (1-\cos\hat{E}) \tag{A.8}$$

then

$$f = 1 - a(1 - \cos\hat{E})/r_o \tag{A.9}$$

$$g = (t-t_o) - a^{3/2}\ (\hat{E} - \sin\hat{E})/\mu^{\frac{1}{2}} \tag{A.10}$$

$$r = a(1 - c_o\ \cos\hat{E}) - D_o\left(\frac{a}{\mu}\right)^{\frac{1}{2}}\sin\ \hat{E} \tag{A.11}$$

$$\dot{f} = -(rr_o)^{-1}\ \mu a\ \sin\hat{E} \tag{A.12}$$

$$\dot{g} = 1 - a(1 - \cos\hat{E})/r \tag{A.13}$$

and finally

$$\begin{Bmatrix} X(t) \\ Y(t) \\ Z(t) \end{Bmatrix} = f \begin{Bmatrix} X_o \\ Y_o \\ Z_o \end{Bmatrix} + g \begin{Bmatrix} \dot{X}_o \\ \dot{Y}_o \\ \dot{Z}_o \end{Bmatrix} \tag{A.14}$$

$$\begin{Bmatrix} \dot{X}(t) \\ \dot{Y}(t) \\ \dot{Z}(t) \end{Bmatrix} = \dot{f} \begin{Bmatrix} X_o \\ Y_o \\ Z_o \end{Bmatrix} + \dot{g} \begin{Bmatrix} \dot{X}_o \\ \dot{Y}_o \\ \dot{Z}_o \end{Bmatrix} \tag{A.15}$$

A.4 Satellite Motion: Transformation to Earth-Fixed Rectangular Coordinates

Regardless of whether the satellite orbit is calculated via the analytical solution of §A.3 or whether perturbations are considered and equations (A.1) are integrated numerically, the same transformations must be applied to the inertial position, velocity, and acceleration coordinates

$$\{X,Y,Z;\dot{X},\dot{Y},\dot{Z};\ddot{X},\ddot{Y},\ddot{Z}\}$$

to obtain the analogous coordinates

$$\{x,y,z;\dot{x},\dot{y},\dot{z};\ddot{x},\ddot{y},\ddot{z}\}$$

with respect to earth-fixed axes. The transformations are compactly written in matrix form as

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} \tag{A.16}$$

$$\begin{Bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{Bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \dot{X} + Y\,\omega_e \\ \dot{Y} - X\,\omega_e \\ \dot{Z} \end{Bmatrix} \tag{A.17}$$

$$\begin{Bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{Bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \ddot{X} + 2\dot{Y}\omega_e - X\omega_e^2 \\ \ddot{Y} - 2\dot{X}\omega_e - Y\omega_e^2 \\ \ddot{Z} \end{Bmatrix} \tag{A.18}$$

where

$$\theta = \theta_o + \omega_e(t-t_o) \tag{A.19}$$

= sidereal time of greenwich

= counter clockwise rotation about the Z=z polar axis

ω_e = angular velocity of the earth

A.5 Satellite Motion: Transformation to Earth-Fixed Ellipsoidal Coordinates

The earth-fixed ellipsoidal coordinates $\{\phi,\lambda,H;\ \dot{\phi},\dot{\lambda},\dot{H};\ \ddot{\phi},\ddot{\lambda},\ddot{H}\}$ can be calculated from the earth-fixed rectangular coordinates $\{x,y,z\ \dot{x},\dot{y},\dot{z};\ddot{x},\ddot{y},\ddot{z}\}$ from the following equations:

Polar Distance and Derivatives

$$r_p = \sqrt{x^2 + y^2} \tag{A.20a}$$

$$\dot{r}_p = (x\dot{x} + y\dot{y})/r_p \tag{A.20b}$$

$$\ddot{r}_p = (x\ddot{x} + y\ddot{y} + \dot{x}^2 + \dot{y}^2 - \dot{r}_p^2)/r_p \tag{A.20c}$$

Longitude (λ) and Derivatives

$$\lambda = \tan^{-1}(y/x) \tag{A.21a}$$

$$\dot{\lambda} = (x\dot{y} - y\dot{x})/r_p^2 \tag{A.21b}$$

$$\ddot{\lambda} = (x\ddot{y} - y\ddot{x} + \dot{x}^2 - \dot{y}^2)/r_p^2 - 2(x\dot{y} - y\dot{x})\dot{r}_p^2/r_p^3 \tag{A.21c}$$

Lattitude (ϕ) and Height (H) and Their Derivatives

ϕ and H are determined via the Newton iteration

$$\begin{Bmatrix} \phi \\ H \end{Bmatrix}^{(k+1)} = \begin{Bmatrix} \phi \\ H \end{Bmatrix}^{(k)} + D^{-1}(k)\begin{Bmatrix} z - z^{(k)} \\ r_p - r_p(k) \end{Bmatrix} \tag{A.22a}$$

$$\begin{Bmatrix} \dot{\phi} \\ \dot{H} \end{Bmatrix} = D^{-1} \begin{Bmatrix} \dot{z} \\ \dot{r}_p \end{Bmatrix} \tag{A.22b}$$

$$\begin{Bmatrix} \ddot{\phi} \\ \\ \ddot{H} \end{Bmatrix} = D^{-1} \begin{Bmatrix} \ddot{z} + \dot{\phi}^2[(\frac{b^2}{a^2}) N + H)\sin\phi - 2\frac{b^2}{a^2}\frac{dN}{d\phi}\cos\phi - \frac{b^2}{a^2}\frac{d^2N}{d\phi^2}\sin\phi]-2\dot{\phi}\dot{H}\cos\phi \\ \\ \ddot{r}_p + \dot{\phi}^2[(N+H)\cos\phi + 2\frac{dN}{d\phi}\sin\phi - \frac{d^2N}{d\phi^2}\cos\phi] + 2\dot{\phi}\dot{H}\sin\phi \end{Bmatrix} \tag{A.22c}$$

where the iteration (A.22a) normally converges in 3 iterations, using the spherical earth starting approximations

$$\phi^{(0)} = \sin^{-1}(\frac{z}{r}) \;,\quad H^{(0)} = r - 6378 \text{ km}$$

and

$$D = D(\phi,H) \equiv \begin{bmatrix} \frac{\partial z}{\partial \phi} & \frac{\partial z}{\partial H} \\ \\ \frac{\partial r_p}{\partial \phi} & \frac{\partial r_p}{\partial H} \end{bmatrix} = \begin{bmatrix} [\frac{b^2}{a^2}\frac{dN}{d\phi}\sin\phi + (\frac{b^2}{a^2} N+H)\cos\phi] & [\sin\phi] \\ \\ [\frac{dN}{d\phi}\cos\phi - (N+H)\sin\phi] & [\cos\phi] \end{bmatrix} \tag{A.23}$$

$$D(k) \equiv D(\phi^{(k)}, H^{(k)}) = \text{eqn(A.23) evaluated with } \phi^{(k)}, H^{(k)}. \tag{A.24}$$

$N = a^2(a^2\cos^2\phi + b^2\sin^2\phi)^{-\frac{1}{2}}$ = radius of curvature in the prime vertical plane (A.25a)

$$\frac{dN}{d\phi} = -\frac{(a^2-b^2)}{2a^4} N^3\sin 2\phi \tag{A.25b}$$

$$\frac{d^2N}{d\phi^2} = -\frac{(a^2-b^2)}{2a^4}[3N^2\frac{dN}{d\phi}\sin 2\phi + 2N^3\cos 2\phi] \tag{A.25c}$$

$$z = (\frac{b^2}{a^2} N + H)\sin\phi \tag{A.26}$$

$$r_p = (N + H)\cos\phi \tag{A.27}$$

$$z^{(k)} = \text{eqn(A.26)} = \text{evaluated with } \phi^{(k)},\ H^{(k)}$$

$r_p^{(k)}$ = eqn (A.27) evaluated with $\phi^{(k)}, H^{(k)}$.

In several instances, the orbit normal unit vector $\hat{\underline{w}}'$ is required in carrying out calculations needed in the text of this report. The scan vector can be determined by the cross product

$$\hat{\underline{w}}' = \underline{r} \times \dot{\underline{r}} / |\underline{r} \times \dot{\underline{r}}| = (h_x/h)\,\underline{I} + (h_y/h)\underline{J} + (h_z/h)\underline{K} \tag{A.28}$$

where

$$h_x = (Y\dot{Z} - Z\dot{Y}) \quad h_y = (Z\dot{X} - X\dot{Z}) \quad , \quad h_z = (X\dot{Y} - Y\dot{X}) \tag{A.29}$$

$$h^2 = h_x^{\,2} + h_y^{\,2} + h_z^{\,2} \tag{A.30}$$

The components (A.29) are constant for Keplerian orbits, but must be calculated instantaneously in the presence of perturbations. The rotating earth-fixed components of $\hat{\underline{w}}'$ are determined via the transformation

$$\begin{Bmatrix} \hat{w}'_x \\ \hat{w}'_y \\ \hat{w}'_z \end{Bmatrix} = \frac{1}{h} \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} h_x \\ h_y \\ h_z \end{Bmatrix} \tag{A.31}$$

and, when necessary, the tangent ($\hat{\underline{t}}$) crosstrack ($\hat{\underline{c}}$) and normal ($\hat{\underline{n}}$) components of $\hat{\underline{w}}$ are determined by

$$\begin{Bmatrix} \hat{w}'_t \\ \hat{w}'_c \\ \hat{w}'_n \end{Bmatrix} = \begin{bmatrix} t_x & t_y & t_z \\ c_x & c_y & c_z \\ n_x & n_y & n_z \end{bmatrix} \begin{Bmatrix} \hat{w}'_x \\ \hat{w}'_y \\ \hat{w}'_z \end{Bmatrix} \tag{A.32}$$

where the $\hat{\underline{t}}$, $\hat{\underline{c}}$, $\hat{\underline{n}}$ vectors are defined (and expression for computing their components are given) in Appendix B(eqn B.8).

APPENDIX B

SUB-POINT GEOMETRY AND MOTION

B.1 The Sub-point Vector $\underline{R}_g$ and its Derivatives

With reference to Figure 6.4b, it is clear that the sub-point vector $\underline{R}_g$ and its earth-fixed derivatives are

$$\underline{R}_g = \underline{r} - \underline{H} = (x - H_x)\,\underline{i} + (y - H_y)\underline{j} + (z - H_z)\underline{k} \tag{B.1a}$$

$$^e\dot{\underline{R}}_g = {}^e\dot{\underline{r}} - {}^e\dot{\underline{H}} = (\dot{x} - \dot{H}_x)\underline{i} + (\dot{y} - \dot{H}_y)\underline{j} + (\dot{z} - \dot{H}_z)\underline{k} \tag{B.1b}$$

$$^e\ddot{\underline{R}}_g = {}^e\ddot{\underline{r}} - {}^e\ddot{\underline{H}} = (\ddot{x} - \ddot{H}_x)\underline{i} + (\ddot{y} - \ddot{H}_z)\underline{j} + (\ddot{z} - \ddot{H}_z)\underline{k} \tag{B.c}$$

The left superscript e denotes that the derivatives of the groundtrack (sub-point) vector $\underline{R}_g$ are taken with respect to earth-fixed axes. The earth-fixed satellite coordinates $(x,y,z;\dot{x},\dot{y},\dot{z};\ddot{x},\ddot{y},\ddot{z})$ are available from Appendix A, eqns (A.16), (A.17) and (A.18). The $\underline{H}$ vector and its derivatives follow from the geometry of Fig. 6.4a as

$$H_x = \mathrm{H}\cos\phi\cos\lambda \equiv \mathrm{H}c\phi c\lambda \tag{B.2a}$$

$$H_y = \mathrm{H}c\phi s\lambda \tag{B.2b}$$

$$H_z = \mathrm{H}s\phi \tag{B.2c}$$

$$\dot{H}_x = \dot{\mathrm{H}}c\phi c\lambda - \mathrm{H}\dot{\phi}s\phi c\lambda - \mathrm{H}\dot{\lambda}c\phi s\lambda \tag{B.3a}$$

$$\dot{H}_y = \dot{\mathrm{H}}c\phi s\lambda - \mathrm{H}\dot{\phi}s\phi s\lambda + \mathrm{H}\dot{\lambda}c\phi c\lambda \tag{B.3b}$$

$$\dot{H}_z = \dot{\mathrm{H}}s\phi + \mathrm{H}\dot{\phi}c\phi \tag{B.3c}$$

$$\begin{aligned}\ddot{H}_x &= \ddot{\mathrm{H}}c\phi c\lambda - \mathrm{H}\ddot{\phi}s\phi c\lambda - \mathrm{H}\ddot{\lambda}s\phi s\lambda - \mathrm{H}(\dot{\phi}^2+\dot{\lambda}^2)c\phi c\lambda \\ &\quad - 2\dot{\mathrm{H}}\dot{\phi}s\phi c\lambda - 2\dot{\mathrm{H}}\dot{\lambda}c\phi s\lambda + 2\mathrm{H}\dot{\phi}\dot{\lambda}s\phi s\lambda\end{aligned} \tag{B.4a}$$

$$\begin{aligned}\ddot{H}_y &= \ddot{\mathrm{H}}\,c\phi s\lambda - \mathrm{H}\ddot{\phi}s\phi s\lambda + \mathrm{H}\ddot{\lambda}c\phi s\lambda - \mathrm{H}(\dot{\phi}^2 + \dot{\lambda}^2)c\phi s\lambda \\ &\quad - 2\dot{\mathrm{H}}\dot{\phi}s\phi s\lambda + 2\dot{\mathrm{H}}\dot{\lambda}c\phi c\lambda - 2\mathrm{H}\dot{\phi}\dot{\lambda}s\phi c\lambda\end{aligned} \tag{B.4b}$$

$$\ddot{H}_z = \ddot{\mathrm{H}}s\phi + \mathrm{H}\ddot{\phi}c\phi + 2\dot{\mathrm{H}}\dot{\phi}c\phi - \mathrm{H}\dot{\phi}^2s\phi \tag{B.4c}$$

B.2 Tangent, Crosstrack and Normal Vectors

Referring to figure 6.4b, the unit vector tangent to the sub-point path generated in earth-fixed axes is clearly ,

$$\hat{\underline{t}} = \frac{1}{V} \, {}^{e}\dot{\underline{R}} = \left(\frac{\dot{x} - \dot{H}_x}{V}\right) \underline{i} + \left(\frac{\dot{y} - \dot{H}_y}{V}\right) \underline{j} + \left(\frac{\dot{z} - \dot{H}_z}{V}\right) \underline{k} \qquad (B.5)$$

Referring to Figure 6.4a the unit vector normal to the ellipsoid is clearly

$$\hat{\underline{n}} = (\cos\phi_g \ \cos\lambda_g)\underline{i} + (\cos\phi_g \sin\lambda_g)\underline{j} + (\sin\phi_g)\underline{k} \qquad (B.6)$$

The crosstrack unit vector is defined according to the right hand rule as

$$\hat{\underline{c}} = \hat{\underline{n}} \times \hat{\underline{t}} \qquad (B.7)$$

In summary

$$\begin{Bmatrix} \hat{\underline{t}} \\ \hat{\underline{c}} \\ \hat{\underline{n}} \end{Bmatrix} = \begin{bmatrix} t_x & t_y & t_z \\ c_x & c_y & c_z \\ n_x & n_y & n_z \end{bmatrix} \begin{Bmatrix} \underline{i} \\ \underline{j} \\ \underline{k} \end{Bmatrix} = \begin{bmatrix} \left(\frac{\dot{x}-\dot{H}_z}{V}\right) & \left(\frac{\dot{y}-\dot{H}_y}{V}\right) & \left(\frac{\dot{z}-\dot{h}_z}{V}\right) \\ (n_y t_z - n_z t_y) & (n_z t_x - n_x t_z) & (n_x t_y - n_y t_x) \\ c\phi_g c\lambda_g & c\phi_g s\lambda_g & s\phi_g \end{bmatrix} \begin{Bmatrix} \underline{i} \\ \underline{j} \\ \underline{k} \end{Bmatrix} \qquad (B.8)$$

The ellipsoid coordinates $\{\phi,\lambda,H\}$ and their derivatives needed in eqns (B.2), (B.3) and (B.4) are determined in terms of satellite motion by equations (A.20), (A.21) and (A.22). Thus the vectors $\underline{R}_g$, ${}^{e}\dot{\underline{R}}_g$, and ${}^{e}\ddot{\underline{R}}_g$ characterizing the sub-point's position, velocity and acceleration can now be calculated from equations (B.1).

B.3 Radius of Curvature of the Sub-Point Path's Projection in the Osculating Tangent Plane.

With reference to Figure 6.3, define

$\rho = \rho(t)$ = instantaneous radius of curvature of the groundtrack's instantaneous projection into the ellipsoid's tangent plane

or

$$\frac{1}{\rho} = \hat{\underline{c}} \cdot \frac{{}^{e}d^2\underline{R}_g}{ds^2} \tag{B.9}$$

= tangent plane component of groundtrack (sub-point) acceleration with respect to earth-fixed axes

where

$\hat{\underline{c}}$ = unit vector normal to groundtrack, lies in the tangent plane

$$= \hat{\underline{n}} \times \hat{\underline{t}} \tag{B.10}$$

$\hat{\underline{t}}$ = unit vector tangent to the groundtrack

$$= \frac{1}{V}\, {}^{e}\dot{\underline{R}}_g = \frac{1}{V}\left[(\dot{x} - \dot{H}_x)\underline{i} + (\dot{y} - \dot{H}_y)\underline{j} + (\dot{z} - \dot{H}_z)\underline{k}\right] \tag{B.11}$$

$\hat{\underline{n}}$ = unit vector normal to the ellipsoid

$$= (c\phi c\lambda)\underline{i} + (c\phi s\lambda)\underline{j} + s\phi)\underline{k} \tag{B.12}$$

= speed of the sub-point with respect to the earth-fixed axes

$$V = |{}^{e}\dot{\underline{R}}_g| = \frac{ds}{dt} = \left[(\dot{x} - \dot{H}_x)^2 + (\dot{y} - \dot{H}_y)^2 + (\dot{z} - \dot{H}_z)^2\right]^{\frac{1}{2}} \tag{B.13}$$

s = arc length along the groundtrack

$$= \int_0^t V(\tau)\, d\tau \tag{B.14}$$

$$\frac{{}^{e}d\underline{R}_g}{ds} = \frac{{}^{e}d\underline{R}_g}{dt} \cdot \frac{dt}{ds} = \frac{1}{V}\, {}^{e}\dot{\underline{R}}_g \tag{B.15}$$

$$\frac{{}^{e}d^2\underline{R}_g}{ds^2} = -\frac{1}{V^2}\frac{dV}{ds}\, {}^{e}\dot{\underline{R}}_g + \frac{1}{V^2}\, {}^{e}\ddot{\underline{R}}_g$$

$$= -\frac{\dot{V}}{V^3}\, {}^{e}\dot{\underline{R}}_g + \frac{1}{V^2}\, {}^{e}\ddot{\underline{R}}_g \tag{B.16}$$

Substitution of Equations (B.1) and (B.16) and then equations (B.11) (B.12) into (B.10) and (B.9) ultimately reduces (B.9) to the explicit formula

$$\frac{1}{\rho(t)} = \frac{1}{V^3}\{(\ddot{x} - \ddot{H}_x)[(\dot{z} - \dot{H}_z)c\phi s\lambda - (\dot{y} - \dot{H}_y)s\phi]$$
$$+ (\ddot{y} - \ddot{H}_y)[(\dot{x} - \dot{H}_x)s\phi - (\dot{z} - \dot{H}_z)c\phi c\lambda]$$
$$+ (\ddot{z} - \ddot{H}_z)[(\dot{y} - \dot{H}_y)c\phi c\lambda - (\dot{x} - \dot{H}_x)c\phi s\lambda]\} \tag{B.17}$$

Equations (B.13) and (B.17) provide the important equations for calculating the sub-point velocity and the sub-point path radius of curvature.

APPENDIX C

4 CYCLE RUNGE-KUTTA ALGORITHM

Given a system of first order ordinary differential equations of the form

$$\frac{dx_i}{dt} = f_i(t,x_1,x_2, \ . \ . \ . \ , x_n) \ , \ i=1,2 \ . \ . \ . \ ,n \qquad (C.1)$$

and specified initial conditions

$$\{x_1(t_o) \ , \ . \ . \ . \ \ x_n(t_o)\} \ , \qquad (C.2)$$

the following 4 cycle Runge-Kutta algorithm permits recursive, step-by-step integration of equations (C.1) to determine the x_i at sequence of times $t_1,t_2, \ . \ . \ . \ .t_k$:

$$x_i(t_{k+1}) = x_i(t_k) + \frac{1}{6}[\Delta_1 x_i + 2\Delta_2 x_i + 2\Delta_3 x_i + \Delta_4 x_i]$$
$$i = 1,2,...,n \qquad (C.3)$$

where

$$\Delta_1 x_i = \Delta t[f_i(t_k . x_1(t_k), \ . \ . \ ., x_n(t_k)], \ i = 1,2, \ . \ . \ ., n \qquad (C.4a)$$

$$\Delta_2 x_i = \Delta t[f_i(t_k + \frac{\Delta t}{2}, \ x_1(t_k) + \frac{\Delta_1 x_1}{2} \ , \ . \ . \ ., x_n(t_k) + \frac{\Delta_1 x_n}{2})]i=1,2,..,n \qquad (C.4b)$$

$$\Delta_3 x_i = \Delta t[f_i(t_k + \frac{\Delta t}{2} \ , \ x_1(t_k) + \frac{\Delta_2 x_1}{2} \ , \ x_n(t_k) + \frac{\Delta_2 x_n}{2} \ )], \ i=1,2...,n \qquad (C.4c)$$

$$\Delta_4 x_i = \Delta t[f_i(t_{k+1}, \ x_1(t_k) + \Delta_3 x_1, \ .., \ x_n(t_k) + \Delta_3 x_n)], \ i=1,2,...n \qquad (c.4d)$$

$$t_{k+1} = t_k + \Delta t \qquad (C.4d)$$

The *step size* Δt must be determined empirically to maintain the desired number of significant figures.

The above Runge-Kutta algorithm is implemented in subroutine RUNGE. For each set of differential equations of the form (C,1), the functions on the right hand side must be programmed in subroutine DERIV.

The subroutines are given in Appendix D, set up for the integrations of §4.1.1, eqns. (11).

APPENDIX D

IMPLEMENTATION AND DOCUMENTATION OF COMPUTER PROGRAMS

The various programs developed in the course of this work can be divided into four major categories, corresponding to the four major functions of this software. The first group (D.1 through D.12) consists of the subroutines necessary to compute the coefficients for the Fourier series fit to the satellite groundtrack projection and the f-function*. The second group (viz., Forward Transformation), subroutines D.13 through D.16, generate map plane x and y coordinates given ϕ and λ. The third group (viz. Inverse Transformation), subroutine D.18, generates ϕ and λ given map plane x and y coordinate. And the last group (viz. sensitivity ananlysis), subroutine D.17, determines length distortions along lines of constant ϕ and λ. In addition to the subroutines cited, the final three groups of subroutines, rely upon several of the first group to perform various secondary calculations.

D.1 Subroutine ROC (T, cons)

This subroutine computes the radius of curvature (eqn. B.17 Appendix B) and the first and second derivatives of the motion along the satellite groundtrack (eqns. A.16 and A.22, Appendix A). Time is input as T. The vector CONS(30) contains the various required constants and initial conditions defining the orbit and the reference ellipsoid. The output of this routine is passed through common blocks and CONS. The calculations are based on the equations of Appendix A and B and the Fortran names in terms of these Appendices are:

---

* $f(t) = \int_{t_0}^{t} \frac{V(\tau)}{\rho(\tau)} d\tau$

$\text{TERM} = \frac{1}{\rho}$

$N(3) = \hat{\underline{n}}$

$\text{DRDS}(3) = \hat{\underline{t}} = \frac{1}{V}\, {}^{e}\dot{\underline{R}}$

$\text{CC}(3) = \hat{\underline{c}} = \hat{\underline{n}} \times \hat{\underline{t}}$

$\text{DRDT}(3) = \dot{\underline{r}} - \dot{\underline{H}}$, $\underline{r}$ = satellite position vector

$\text{DDRDT}(3) = \ddot{\underline{r}} - \ddot{\underline{H}}$, $\underline{H}$ = height above the surface of the earth

$\text{PHI} = \phi_g$

$\text{LAMDA} = \lambda_g$

$\text{DSDT} = |{}^{e}\dot{\underline{R}}_g| = V$

$\text{DDSDT} = \dot{V}$

$\text{DPHI} = \dot{\phi}_g$

$\text{DLAMDA} = \dot{\lambda}_g$

$\text{XN}(3) = X, Y, Z$

$\text{DXN}(3) = \dot{X}, \dot{Y}, \dot{Z}$ required for scan vector calculations

$\text{DDXW}(3) = \ddot{X}, \ddot{Y}, \ddot{Z}$

Subroutine ROC has external references to ORBIT, ROTATE, CROSS, EFRAME, PHIH, DPHIDH, and VECPRD.

D.2 Subroutine ORBIT (X,XF,TI,TF,CONS)

This subroutine uses the $f, g, \dot{f}$, and $\dot{g}$ solution of Appendix A to compute the orbit state at time TF for all species of elliptical orbits (including circular). X is the initial state vector at time TI and XF is the state vector at time TF. CONS is the vector used to pass various constants and the output of subroutine ROC, The position,

velocity and time are given in units of km, km/sec and sec respectively. Subroutine ORBIT has an external reference to NEWTON.

D.3 Subroutine NEWTON (CONS, PHI, TI, TF)

This subroutine uses Newton's method to iteratively solve Kepler's equation for change in eccentric anomaly $\hat{E}$ (eqn. A.8 Appendix A).

Subroutines ORBIT and NEWTON provide the means to compute the satellite position at arbitrary given times as a function of the given initial conditions.

D.4 Subroutine ROTATE (M,TO,T,WE)

This subroutine computes the earth's rotation direction cosine matrix M(3,3). TO is the initial time. T is the final time. And WE is the earth's rotation rate. θ_o is passed in CONS.

D.5 Subroutine CROSS (A,B,C)

This subroutine computes the cross product of two vectors (e.g. $\underline{A} \times \underline{B} = \underline{C}$).

D.6 Subroutine EFRAME (A,B,N)

This subroutine rotates given components of an arbitrary vector from the inertial ref. frame to the instantaneous earth-fixed, equatorial frame. A(3) is the vector to be operated upon, B(3,3) is the rotation matrix, and N is dimension of A. The output overwrites the input and is returned in A(3).

D.7 Subroutine PHIH (RXY,Z,RN,PHI,H,LAMDA,A,B)

This subroutine uses a 2-dimensional Newton's method to find phi (geodetic lattitude) and H (the height of the satellite above the reference ellipsoid surface). The constants and corresponding definition is:

$Z = z$

$RXY = (x^2 + y^2)^{\frac{1}{2}}$

$RN = (x^2 + y^2 + z^2)^{\frac{1}{2}}$

A = earth's semi-major axis

B = earth's semi-minor axis

X,Y,Z = Earth fixed components of the satellite position(x,y,z)

See equation A.22a in Appendix A.

D.8 Subroutine DPHIDH(A,B.H.HP.HPP.PHI,DPHI,DDPHI.X.RXY.R,MU,DRDT,DDRDT)

This subprogram computes the first and second derivatives of PHI (geodetic lattitude) and H (the height of the satellite above the ellipsoid surface) (see Appendix A eqn. A.22b and A.22c). X(3) is the vector containing the earth fixed components of the satellite position. The Fortran definition of the output are:

$HP = \dot{H}$

$HHP = \ddot{H}$

$DPHI = \dot{\phi}$

$DDPHI = \ddot{\phi}$

D.9 Subroutine VECPRD (A,B,C.NU)

This subroutine computes the inner product two vectors. If NU $\neq 0$ then $C = \underline{A} \cdot \underline{B}$. If NU = 0 then $C = \underline{A} \cdot \underline{A}$.

D.10 Subroutine INTEG(Z,CONS,TO,TF)

This subroutine integrates to determine the satellite groundtrack map plane coordinates as well as the coefficient for the Fourier series representation of these coordinates.

Z(40) is the vector containing the state at each step in the

integration, CONS(30) is the vector containing initial conditions and constants. TO is the lower limit of integration and TF is the upper limit of integration. The components of the Z vector are defined as follows:

$$Z_1(t) = x_g(t) = \int_{to}^{t} V(\tau)\cos(f(\tau))d\tau$$

$$Z_2(t) = y_g(t) = y_{g(0)} + \int_{to}^{t} V(\tau)\sin(f(\tau))d\tau$$

$$Z_3(t) = f(t) = \int_{to}^{t} \frac{V(\tau)}{\rho(\tau)} d\tau$$

$$Z_4(t) = A_1(t) = \frac{1}{P}\int_{to}^{t} V(\tau)\cos(f(\tau))d\tau$$

$$Z_{n+3}(t) = A_n(t) = \frac{2}{P}\int_{to}^{t} [x_g(\tau) - \bar{\dot{x}}_g\tau]\sin(\frac{n\pi\tau}{P})d\tau$$

$$n=1,2-9,\ 10$$

$$Z_{14}(t) = B_1(t) = \frac{1}{2}\ \frac{1}{P}\int_{to}^{P} y_g(\tau)d\tau$$

$$Z_{n+13}(t) = B_n(t) = \frac{2}{P}\int_{to}^{t} y_g(\tau)\cos(\frac{n\pi\tau}{P})d\tau$$

$$Z_{n+23} = C_n(t) = \frac{2}{P}\int_{to}^{t} f(\tau)\sin(\frac{n\pi\tau}{P})d\tau$$

In this program one preliminary integration is performed to determine $\bar{\dot{x}}_g$ and $\bar{y}_{og}$. Note $\bar{y}_{og} = B_1$ although, it is not used in the Fourier series representation of $y_g(t)$ (so that $y_g(t)$ will oscillate about the x axis with zero mean).

The sole output of these subroutines consists of the Fourier fit coefficients.

Subroutine INTEG has an external reference to RUNGE.

D.11 Subroutine RUNGE(TK,N,DELT,CONST,XK)

This subroutine uses a 4-cycle Runge-Kutta algorithm to approximate the integration of the state XK over the interval [TK,TK+DELT], as described in Appendix C,

i.e. $$\underline{XK}(TK + DELT) = \underline{XK}\ (TK) + \int_{TK}^{TK + DELT} \left[\frac{d}{d\tau}\ \underline{XK}(\tau)\right]d\tau,$$

N is the dimension of the current state vector. Const(30) is the ubiquitious vector containing the necessary constants!

Subroutine RUNGE has external references to ROC and DERIV.

D.12 Subroutine DERIV(T,N,X,CONST,F)

This subroutine computes the derivative approximations required in subroutine RUNGE. T is the current time. N is the dimension of the state vector X. CONST(30) is the vector of necessary constants, And F(N) contains, as output, the derivative approximations.

With subroutine DERIV, we complete the section for determining the satellite ground map plane coordinates.

D.13 Subroutine NS(CONS,PHI,LAMDA,ALPHAO,XT,YT,TIME,NU)

This subroutine computes the x and y map plane coordinates, given ellipsoid coordinates PHI and LAMDA. CONS is the vector containing necessary constants. ALPHAO is the initial equatorial right ascension displacement angle of the spacecraft relative to the inertial X axis (usually taken along the intersection of the initial greenwich meridian plane and the equatorial plane). TIME is time t* as found in NS. If NU≠ 100 seach for t*. If NU=100 bypass the t* time search, XT and YT are the map plane coordinates.

Subroutine NS has external references to SETCON, ROC, ROTATE, CROSS, EFRAME, and FNTS2.

D.14 Subroutine SETCON (A,B,PIE,PERIOD,CONS)

This subroutine sets parameter values not passed in CONS. A is is the reference ellipsoid semi-major axis, and B is the semi-minor axis in km. PERIOD is the satellite orbital period in seconds. CONS is vector of necessary constants.

D.15 Subroutine FNTS2 (T,CONS,XCOEFFS,YCOEFFS, FCOEFFS, F, NX, NY, NZ)

This subroutine generates values for $x_g, y_g, f, \frac{dx_g}{dt}, \frac{dy_g}{dt}$, and $\frac{df}{dt}$ by the Fourier series representations and the differential eqns. for x_g, y_g and f. T is the current time. CONS is the vector of necessary constants. XCOEFFS, YCOEFFS and FCOEFFS are the Fourier series coefficients. NX, NY and NF are the number of coefficients for x_g, y_g, and f respectively. As output

$F(1) = x_g$

$F(2) = y_g$

$F(3) = \gamma(\underline{\hat{t}}, \underline{\hat{c}}$ - Frame rotation angle)

$F(4) = f$

$DX = \frac{dx_g}{dt}$

$DY = \frac{dy_g}{dt}$

$DF = \frac{df}{dt}$.

Subroutine FNTS2 has external references to SETCON and ANG.

D.16 Subroutine ANG(C,S,T,N)

This subroutine computes multiple angle sine and cosine terms for

the Fourier series for x_g, y_g and f. C is the vector containing the cosine (NT) terms. S is the vector containing the sine (NT) terms. N is the number of terms desired, and T is the time which has been normalized to T = π (real time)/(satellite period).

This completes the section for determining the forward transformation.

D.17 Subroutine ERRORS(A,B,CONS,FF,HP,K,U1,U2, V1, V2, ANG1, ANG2, T)

This subroutine computes the partial derivatives

$$U1 = \frac{\partial x}{\partial \phi}$$

$$U2 = \frac{\partial x}{\partial \lambda}$$

$$V1 = \frac{\partial y}{\partial \phi}$$

$$V2 = \frac{\partial y}{\partial x}$$

and the length distortion factors HP and K. HP is $(\frac{\partial s'}{\partial s})$ for lines of contrast λ and K is $(\frac{\partial s'}{\partial s})$ for lines of constant ϕ.

A = earth semi-major axis

B = earth semi-minor axis

FF = flattening factor

ANG1 = lamda of current point

ANG2 = phi of current point

T = current time (as found in NS or INVERSE)

Subroutine ERRORS has external references to ROC, ROTATE, CROSS, VECPRD, and FNTS2.

This completes the section on sensitivity analysis

D.18 Subroutine INVERSE (TO,TI,CONS,FF,X,Y,PHI,LAMDA)

This subroutine computes PHI and LAMDA given map projection plane x and y coordinates. TO is the initial time. TI is the current time.

FF is the flattening factor.

Subroutine INVERSE has external references to SETCON, ROC, ROTATE, CROSS, EFRAME, FNTS2, NS, and ERRORS.

This completes the section for the inverse transformation.

D.19 Subroutine Scanner (T, R1, R2, R3, X1, X2, X3, N1, N2, N3, H, DR1, DR2, DR3, C1, C2, C3)

This program computes a linear correction to time when the scan speed cannot be considered infinite.

SPACE OBLIQUE MERCATOR

PROGRAM LISTING

(Programmed by James D. Turner)

The subroutine listings follow in the following order:

```
      PROGRAM     SOM(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE10,TAPE12
     1,TAPE14,TAPE15,TAPE35)
C
C*****************************************************************************
C         THIS PROGRAM ACTS AS THE DRIVER FOR THE S.O.M. MAP PROJECTION
C         SUBROUTINES.   THE MODE OF OPERATION IS CONTROLLED BY THE INPUT
C         VARIABLE NUMBER.
C
C         IF NUMBER = 0 COMPUTE THE FOURIER COEIFFICIENTS TO THE S.G.T.
C         IF NUMBER = 1 USE THE FORWARD TRANSFORMATION MODE
C         IF NUMBER = 2 USE THE INVERSE TRANSFORMATION MODE
C         IF NUMBER = 3 USE THE SENSITIVITY ANALYSIS MODE
C
C*****************************************************************************
C
      COMMON/DXYMAP/DX,DY
      COMMON/NSDATA/DDRDT,DRDT,C,H,XX,N
      COMMON/ROCDATA/A,B,XO,TO
      COMMON/XYCOEFFS/XCOEFFS,YCOEFFS,FCOEFFS,NX,NY,NF
      COMMON/LE/R
      DIMENSION XCOEFFS(20),YCOEFFS(20),FCOEFFS(20),F(4)
      DIMENSION XO(6),CONS(30),Z(40),GT(3),R(3)
      DIMENSION DDRDT(3),DRDT(3),C(3),XX(6),N(3)
      REAL K
      REAL N,LAMDA
      CALL SETUP(XO,TO,T,CONS)
      CALL SETCON(A,B,PIE,P,CONS)
      CALL ROC(0.,CONS)
      DO 70 I=1,3
   70 CONS(I + 22)=XX(I) - H*N(I)
      FF=CONS(7)
      ESQ=2.*FF - FF*FF
      A=CONS(8)
      B=CONS(9)
      WRITE(6,6000)
      WRITE(6,6005)FF,A,B
 6000 FORMAT(///49X,* EARTH RELATED PARAMETERS*//)
 6005 FORMAT(44X,* FLATTENING FACTOR=*,F15.5,/44X,* SEMI-MAJOR=*,E22.5,
     1/44X,* SEMI-MINOR AXIS=*,E17.5//)
      RADEG=PIE/180.
      READ(5,4)NUMBER
      WRITE(6,4)NUMBER
    4 FORMAT(I2)
      IF(NUMBER.EQ.0)GO TO 1000
      READ(10,1)M,NX
      READ(10,2)(XCOEFFS(I),I=1,NX)
      READ(12,1)M,NY
      READ(12,2)(YCOEFFS(I),I=1,NY)
      READ(14,1)M,NF
      READ(14,2)(FCOEFFS(I),I=1,NF)
    1 FORMAT(2I5)
    2 FORMAT(5E16.10)
      WRITE(6,6010)
 6010 FORMAT(///43X,* COEFFICIENTS FOR THE FOURIER FIT XG,YG AND F*//)
      WRITE(6,6015)
 6015  FORMAT(31X,* X-COEFFS.*,20X,* Y-COEFFS.*,20X,* F-COEFFS.*//)
      WRITE(6,6020)((XCOEFFS(I),YCOEFFS(I),FCOEFFS(I)),I=1,NX)
```

```
 6020 FORMAT(24X,E18.7,13X,E18.7,13X,E18.7)
      IF(NUMBER.EQ.1)GO TO 1010
      IF(NUMBER.EQ.2)GO TO 1020
      IF(NUMBER.EQ.3)GO TO 1030
 1000 CONTINUE
      TF=TO + 2
      CALL INTEG(Z,CONS,TO,TF)
      M=401
      WRITE(10,1)M,NX
      WRITE(10,2)(XCOEFFS(I),I=1,NX)
      WRITE(12,1)M,NY
      WRITE(12,2)(YCOEFFS(I),I=1,NY)
      WRITE(14,1)M,NF
      WRITE(14,2)(FCOEFFS(I),I=1,NF)
      WRITE(6,6010)
      WRITE(6,6015)
      WRITE(6,6020)((XCOEFFS(I),YCOEFFS(I),FCOEFFS(I)),I=1,NX)
      CALL EXIT
 1010 CONTINUE
      PHI=.1155860033E1
      LAMDA=.2375705986E1
      PHI=-.2888446568E-2
      LAMDA=-.1225761632
C        TEST FORWARD TRANSFORMATION
      WRITE(6,3333)
 3333 FORMAT(1H1)
      WRITE(6,4500)
 4500 FORMAT(///39X,* FORWARD TRANSFORMATION TO FIND X,Y GIVEN PHI,LAMDA
     1    *//)
      WRITE(6,4505)PHI,LAMDA
 4505 FORMAT(55X,* PHI    =*,F12.6,/55X,* LAMDA =*,F12.6//)
      CALL NS(CONS,PHI,LAMDA,0,X,Y,T,0)
      WRITE(6,4510)T
 4510 FORMAT(55X,* T-STAR=*,F12.6//)
      WRITE(6,4512)
 4512 FORMAT(49X,* MAP PROJECTION COORDINATES*//)
      WRITE(6,4515)X,Y
 4515 FORMAT(44X,* X =*,E18.7,* Y =*,E18.7//)
      CALL EXIT
 1020 CONTINUE
      XTEST=.1009966176E5
      YTEST=-.5958680333E2
      YTEST=0.79136262E3
      XTEST=0.377871674 6E5
C        INVERSE TRANSFORMATION
      WRITE(6,4790)
 4790 FORMAT(1H1)
      WRITE(6,4800)
 4800 FORMAT(39X,* INVERSE TRANSFORMATION TO FIND PHI, LAMDA GIVEN X,Y
     1    *//)
      WRITE(6,4805)XTEST,YTEST
 4805 FORMAT(44X,* X=*,E18.7,* Y=*,F18.7//)
      CALL INVERSE(TO,T,CONS,FF,XTEST,YTEST,PHI,LAMDA)
      WRITE(6,4810)PHI,LAMDA
 4810 FORMAT(/39X,* THE (PHI,LAMDA) OF THE MAP PLANE POINT (X,Y)
     1*,//54X,* PHI    =*,E18.7,/54X,* LAMDA =*,E18.7///)
      WRITE(6,4510)T
```

```
      CALL EXIT
 1030 CONTINUE
C         SENSITIVITY ANALYSIS
      WRITE(6,5500)
 5500 FORMAT(1H1,47X,*  SENSITIVITY ANALYSIS  *//       )
      WRITE(6,5505)
 5505 FORMAT(35X,* LENGTH DISTORTIONS FOR PTS. SYMMETRICALLY PLACED ON*/
     1 35X,* BOTH SIDES OF THE SATELLITE GROUND TRACK FOR THE */,
     235X,* DISPLACEMENT INCREMENT DELTA = 55.66 KM.*//)
      DELP=8.
      NDELP=DELP +1.001
      DT=P/DELP
      T=-DT
      DO 300 JJ=1,NDELP
      T=T+DT
      ETA=T
      CALL ROC(T,CONS)
      PHI=CONS(14)
      LAMDA=CONS(15)
      PP=PHI/RADEG
      AL=LAMDA/RADEG
      WRITE(6,375)PP,AL,T
  375 FORMAT(////35X,* PHI,LAMDA OF THE GROUND TRACK*,2F15.5//,
     1 39X,* TIME ALONG THE SATELLITE GROUND TRACK = *,F10.5//)
C         GROUND TRACK
      DO 33 I=1,3
   33 GT(I)=XX(I) - H*N(I)
      S-(PIE/180.)*A*2.
      DELTS=S/4.
      DO 250 II=1,9
      DSN=- S + (II - 1)*DELTS
      XN=DSN*C(1) + GT(1)
      YN=DSN*C(2) + GT(2)
      ZN=DSN*C(3) + GT(3)
      LAMDA=ATAN2(YN,XN)
      RXY=SQRT(XN*XN + YN*YN)
      RN=SQRT(XN*XN + YN*YN + ZN*ZN)
      CALL PHIH(RXY,ZN,RN,PHI,H,LAMDA,A,B)
      CALL NS(CONS,PHI,LAMDA,0.,X,Y,T,0)
      ANG1=LAMDA
      ANG2=PHI
      CALL ERRORS(A,B,CONS,FF,HP,K,U1,U2,V1,V2,ANG1,ANG2,T)
      IF(II.EQ.5)GO TO 5900
      WRITE(6,425)HP,K
  425 FORMAT(37X,2F20.6/)
      GO TO 6050
 5900 CONTINUE
      WRITE(6,4251)HP,K
 4251 FORMAT(37X,2F20.6,*  (SATELLITE GROUND TRACK)* /   )
 6050 CONTINUE
  250 CONTINUE
  300 CONTINUE
      STOP
      END
```

```
      SUBROUTINE NS(CONS,PHI,LAMDA,ALPHAO,XT,YT,TIME,NU)
C
C**************************************************************************
C          THIS PROGRAM COMPUTES X AND Y MAP PLANE COORDINATES GIVEN
C          PHI AND LAMDA.
C
C          IF NU = 100 USE TIME(TI) AS FOUND IN SUB. INVERSE.
C
C          INPUT
C              PHI   = GEODETIC LATTITUDE
C              LAMDA = LONGITUDE
C              TIME  = CURRENT TIME IN THE PROGRAM
C
C          OUTPUT
C              XT    = X-MAP PLANE COORDINATE
C              YT    = Y-MAP PLANE COORDINATE
C**************************************************************************
C
      COMMON/DXYMAP/DX,DY
      COMMON/ANG/ALPHAC
      COMMON/NSDATA/DDRDT,DRDT,C,H,X,N
      COMMON/XYCOEFS/XCOEFFS,YCOEFFS,FCOEFFS,NX,NY,NF
      COMMON/INERTAL/XN,DXN,DDXN
      COMMON/SCAN/TP,DPDT,W,DWDT,DT,T
      DIMENSION U(3),DU(3),YN(3),DXN(3),DDXN(3),W(3),DW(3),T(3),DT(3)
      DIMENSION DELRDT(3),DC(3),DN(3),DWDT(3),E(4),F(4),ROT(3,3),RI(3)
      COMMON/ROCDATA/A,B,XO,TO
      COMMON/LE/P
      DIMENSION C(3),DDRDT(3),DRDT(3)
      DIMENSION CONS(1),R(3),N(3),X(6),XO(6),DR(3),TP(3)
      DIMENSION E(4),XCOEFFS(20),YCOEFFS(20),FCOEFFS(20)
      REAL M,LAMDA,N
C
      TIME=TI
      CALL SETCON(A,B,PIE,P,CONS)
      C=COS(PHI)
      S=SIN(PHI)
      CL=COS(LAMDA)
      SL=SIN(LAMDA)
      M=A*A/SQRT(A*A*C*C + B*B*S*S)
C
C          COMPUTE THE EARTH FIXED COMPONENTS OF THE POINT TO BE MAPPED
C
      RI(1)=M*C*CL
      RI(2)=M*C*SL
      RI(3)=B*B*M*S/A/A
C
C          GET THE INITIAL GUESS ON TIME
      DO 35 I=1,3
   35 R(I)=RI(I)
      P=CONS(5)
      T1=R(1)*R(1) + R(2)*R(2) + R(3)*R(3)
      T2=R(1)*CONS(23) + R(2)*CONS(24) + R(3)*CONS(25)
      T3=CONS(23)*CONS(23) + CONS(24)*CONS(24) + CONS(25)*CONS(25)
      ALPHA=ACOS(T2/SQRT(T1*T3))
C
      IF(LAMDA.GE.PIE/2..AND.LAMDA.LE.3.*PIE/2.)ALPHA=2.*PIE - ALPHA
```

BEST AVAILABLE COPY

```
      IF(NL.EQ.100)GO TO 27
      TI=ALPHA*P/(2.*PIE)
   27 CONTINUE
C
      ICOUNT=0
      CONS(16)=0.
      TOLD=1.E30
C
    8 CALL ROC(TI,CONS)
C
C
      CALL ROTATE(ROT,TO,TI,WE)
C
C
C
      DO 90 I=1,3
   90 DR(I)=R(I) -(Y(I) - H*N(I))
      DO 451 I=1,3
  451 DW(I)=DR(I)
      V=CONS(18)
      DVDT=CONS(17)
C
C          XN = POSITION IN INERTIAL SPACE
C         DXN = VELOCITY IN INERTIAL SPACE
C        DDXN = ACCELERATION IN INERTIAL SPACE
C
C          U IS THE VECTOR IN THE SCAN DIRECTION
C
      CALL CROSS(XN,DXN,U)
C
C
      CALL CROSS(XN,DDXN,DU)
C
C
C          DU IS THE DERIVATIVE OF THE SCAN VECTOR IN THE E-FRAME WITH COMPONENTS
C          IN INERITAL SPACE
C
      DU(1)=DU(1) + WE*U(2)
      DU(2)=DU(2)  - WE*U(1)
      DU(3)=DU(3)
C
C
C          ROTATE THE COMPONENTS OF U,DU TO THE E-FRAME.
C
      CALL EFRAME(U,ROT,3)
      CALL EFRAME(DU,ROT,3)
C
C          T IS THE UNIT VECTOR ALONG TRACK  WITH COMPONENTS IN THE E-FRAME
      DO 15 I=1,3
      T(I)=DRDT(I)/V
   15 DT(I)=DDRDT(I)/V - DVDT*DRDT(I)/(V*V)
C
C          C IS THE UNIT VECTOR IN THE CROSS TRACK DIRECTION  WITH COMPONENTS IN
C          THE E-FRAME.
C
      CALL CROSS(N,T,C)
      OPHT=CONS(26)
```

BEST AVAILABLE COPY

```
      DLAMDA=CONS(27)
      CGT=COS(CONS(14))
      SGT=SIN(CONS(14))
      CLGT=COS(CONS(15))
      SLGT=SIN(CONS(15))
C
C        DN IS THE DERIVATIVE OF THE NORMAL VECTOR IN THE E-FRAME
C
      DN(1)=-SGT*CLGT*DPHI - CGT*SLGT*DLAMDA
      DN(2)=-SGT*SLGT*DPHI + CGT*CLGT*DLAMDA
      DN(3)=CGT*DPHI
C
      CALL CROSS(DN,T,E)
      CALL CROSS(N,DT,F)
C
C        DC IS THE DERIVATIVE OF THE C VECTOR IN THE E-FRAME.
C
      DO 19 I=1,3
   19 DC(I)=E(I) + F(I)
C
      T1=-(DRDT(1)*T(1) + DRDT(2)*T(2) + DRDT(3)*T(3))
      T2=DR(1)*DT(1) + DR(2)*DT(2) + DR(3)*DT(3)
      T3=DR(1)*T(1) + DR(2)*T(2) + DR(3)*T(3)
      T4=-(DRDT(1)*C(1) + DRDT(2)*C(2) + DRDT(3)*C(3))
      T5=DR(1)*DC(1) + DR(2)*DC(2) + DR(3)*DC(3)
      T6=DR(1)*C(1) + DR(2)*C(2) + DR(3)*C(3)
C
C        DR IS THE PROJECTION OF THE DISPLACEMENT VECTOR ONTO THE T,C PLANE.
C
      DO 339 I=1,3
  339 DR(I)=T3*T(I) + T6*C(I)
C
C        DFLRDT IS THE DERIVATIVE OF THE DISPLACEMENT VECTOR IN THE T,C PLANE
C
      DO 49 I=1,3
   49 DFLRDT(I)=(T1+T2)*T(I) + T3*DT(I) + (T4+T5)*C(I) + T6*DC(I)

      T1=DU(1)*T(1) + DU(2)*T(2) + DU(3)*T(3)
      T2=U(1)*DT(1) + U(2)*DT(2) + U(3)*DT(3)
      T3=U(1)*T(1) + U(2)*T(2) + U(3)*T(3)
      T4=DU(1)*C(1) + DU(2)*C(2) + DU(3)*C(3)
      T5=U(1)*DC(1) + U(2)*DC(2) + U(3)*DC(3)
      T6=U(1)*C(1) + U(2)*C(2) + U(3)*C(3)
C
      TERM=T3*(T1+T2) + T6*(T4+T5)
      SUMSQ=SQRT(T3*T3 + T6*T6)
      AA=T3/SUMSQ
      BB=T6/SUMSQ
C
C        W IS NOW THE SCAN VECTOR AS PROJECTED ONTO THE T,C PLANE
C
      DO 45 I=1,3
   45 W(I)=AA*T(I) + BB*C(I)
C
      DA=(T1+T2)/SUMSQ - T3*TERM/(SUMSQ*SUMSQ*SUMSQ)
      DB=(T4+T5)/SUMSQ - T6*TERM/(SUMSQ*SUMSQ*SUMSQ)
C
```

```
C         DWDT IS THE DERIVATIVE OF THE SCAN VECTOR AS PROJECTED ONTO THE T,C
C         PLANE
C
      DO 51 I=1,3
   51 DWDT(I)=DA*T(I) + AA*DT(I) + DB*C(I) + BB*DC(I)
C
C
      CALL CROSS(W,N,TP)
      CALL CROSS(DWDT,N,E)
      CALL CROSS(W,DN,F)
C
      DO 105 I=1,3
  105 E(I)=E(I) + F(I)

      TERM=DR(1)*TP(1) + DR(2)*TP(2) + DR(3)*TP(3)
      TERM1=DR(1)*E(1) + DR(2)*E(2) + DR(3)*E(3)
      TERM2=DELRDT(1)*TP(1) + DELRDT(2)*TP(2) + DELRDT(3)*TP(3)
C
      FT=TERM
C
      DFDT=TERM1 + TERM2
      IF(NU.EQ.100)GO TO 355
C
      TI=TI - FT/DFDT
C
  355 CONTINUE
      DP=DW(1)*C(1) + DW(2)*C(2) + DW(3)*C(3)
      IF(NU.EQ.100) GO TO 20
      IF(ABS(TI-TOLD).LE.1.E-05) GO TO 20
      TOLD=TI
C
C
      ICOUNT=ICOUNT + 1
      IF(ICOUNT.EQ.8) GO TO 37
      IF(ICOUNT.EQ.8) CONS(16)=1
      IF(CONS(16).EQ.1) RETURN
      GO TO 8
   37 WRITE(6,1000)
 1000 FORMAT(1H1)
      WRITE(6,1001)
 1001 FORMAT(*  FAILED TO CONVERGE IN NS TIME SEARCH*//)
C
      STOP
   20 CONTINUE
      IF(NU.EQ.100)GO TO 411
      CALL SCANNER(TI,RI(1),RI(2),RI(3),X(1),X(2),X(3),N(1),N(2),N(3),
     1             F,DW(1),DW(2),DW(3),C(1),C(2),C(3))
  411 CONTINUE
      TIME=TI
C
      CALL FNTS2(TI,CONS,XCOEFFS,YCOEFFS,FCOEFFS,F,NX,NY,NF)
C
      TT=SQRT(R(1)*R(1) + R(2)*R(2) + R(3)*R(3))

      DO 350 I=1,3
  350 DR(I)=X(I) - F*N(I)
C
```

```
C        RCT IS THE RADIUS OF CURVATURE IN THE CROSS TRACK DIRECTION
C
      RCT=DR(1)*N(1) + DR(2)*N(2) + DR(3)*N(3)*A*A/B/B
C
C        AS ORIGINALLY DEFINED THE RCT VECTOR IS DOWN THE NEG. N-AXIS.
C
      RCT=ABS(RCT)
      ALPHAC=ATAN2(LP,RCT)
      IF(ABS(ALPHAC).GT.(20.*PIE/180.))CONS(16)=1
      IF(CONS(16).EQ.1)RETURN
      TERM=TAN(ALPHAC/2. + PIE/4.)
      TERM2=W(1)*T(1) + W(2)*T(2) + W(3)*T(3)
      THETAS=-ASIN(TERM2)
      D=RCT*ALOG(TERM)/COS(THETAS)
      XT=F(1) + D*COS(F(3) + THETAS)
      YT=F(2) + D*SIN(F(3) + THETAS)
C
      RETURN
      END
```

```
      SUBROUTINE ERRORS(A,B,CONS,FF,HP,K,U1,U2,V1,V2,ANG1,ANG2,TIME)
C
C*****************************************************************************
C        THIS PROGRAM COMPUTES THE PARTIALS DERIVATIVES  REQUIRED TO DETERMINE
C        LENGTH DISTORTIONS IN THE MAP PLANE. THESE PARTIALS ARE
C        ALSO USED BY SUBR. INVERSE TO FIND PHI AND LAMDA GIVEN X AND Y.
C
C        INPUT
C              A  = SEMI-MAJOR AXIS OF THE EARTH
C              B  = SEMI-MINOR AXIS OF THE EARTH
C              FF = FLATTENING FACTOR
C              ANG1 = LAMDA OF THE MAPPING PT. OFF THE GROUND TRACK
C              ANG2 = PHI OF THE MAPPING PT. OFF THE GROUND TRACK
C              TIME = THE CURRENT TIME AS FOUND IN SUBROUTINE NS.
C        OUTPUT
C              PARTIALS
C              U1 = DX/D(PHI)
C              U2 = DX/D(LAMDA)
C              V1 = DY/D(PHI)
C              V2 = DY/D(LAMDA)
C              DISTORTIONS FACTORS
C              HP = LENGTH DISTORTIONS ALONG LINES OF CONST LAMDA
C              K  = LENGTH DISTORTIONS ALONG LINES OF CONST PHI
C*****************************************************************************
C
      COMMON/NSDATA/DDRDT,DRDT,CC,H,X,N
      COMMON/DF/DFT
      COMMON/DXYMAP/DXDT,DYDT
      COMMON/ANG/ALPHAC
      COMMON/DERVIS/TEST(20)
      COMMON/XYCOEFFS/XCOEFFS,YCOEFFS,FCOEFFS,NX,NY,NF
      COMMON/SCAN/W,DFDT,WP,DWPDT,DTDT,TP
      DIMENSION DTDT(3),DTPDPHI(3),DTPDLAM(3),TP(3)
      DIMENSION   RI(3),    X(6),DRDT(3),DDRDT(3),T(3),SS(3),CONS(1)
      DIMENSION DNDPHI(3),DNDLAM(3),DUMMY(3),E(3),F(4),G(3),N(3)
      DIMENSION WP(3),DWPDT(3),DWPDPHI(3),DWPDLAM(3),CC(3)
      DIMENSION XCOEFFS(20),YCOEFFS(20),FCOEFFS(20)      ,R(3),DR(3)
      DIMENSION W(3),DCDPHI(3),DCDLAM(3),DRPHI(3),DRLAM(3),ROT(3,3)
      REAL M,N,LAMDA
      REAL K
      TI=TIME
      CALL ROC(TI,CONS)
      PHI=CONS(14)
      LAMDA=CONS(15)
      V=CONS(18)
      DVDT=CONS(17)
      DPHI=CONS(26)
      DLAMDA=CONS(27)
C        F-FRAME COMPONENTS OF S.G.T.
      RX=X(1) - H*N(1)
      RY=X(2) - H*N(2)
      RZ=X(3) - H*N(3)
      PIE=2.*ASIN(1.)
      CL=COS(ANG1)
      SL=SIN(ANG1)
      C=COS(ANG2)
      S=SIN(ANG2)
```

```
      M=A*A/SQRT(A*A*C*C + B*B*S*S)
C                COMPONENTS OF THE PT. TO BE MAPPED
      R(1)=M*C*CL
      R(2)=M*C*SL
      R(3)=B*B*M*S/A/A
      DO 49 I=1,3
   49 RI(I)=R(I)
      T0=0.
C        WE IS THE ROTATION RATE OF THE EARTH
      CALL ROTATE(RCT,T0,TI,WE)
C        F-FRAME COMPONENTS OF THE PT. TO BE MAPPED
      DO 3 I=1,3
    3 DR(I)=R(I) - (X(I) - H*N(I))
      TANG2=2.*ANG2
C
      DMDPHI=(A*A - B*B)*M*M*M*SIN(TANG2)/(2.*A*A*A*A)
C
      DO 45 I=1,3
   45 DUMMY(I)=DRDT(I)/V
C
C              COMPUTE THE COMPONENTS OF THE C VECTOR
C
      CALL  CROSS(N,DUMMY,CC)
C
C        T(I),I=1,3 ...DR(PHI,LAMDA)/DPHI, DR=R(PHI,LAMDA) - R(I)
C        S(I),I=1,3 ...DR(PHI,LAMDA)/DLAMDA, DR=R(PHI,LAMDA) - R(T)
      T(1)=DMDPHI*C*CL - M*S*CL
      T(2)=DMDPHI*C*SL - M*S*SL
      T(3)=(B*B/(A*A))*(DMDPHI*S + M*C)
C
      SS(1)=-M*C*SL
      SS(2)=M*C*CL
      SS(3)=0.
C
C        F = DR.DRDT   , WHERE  (.)= INNER PRODUCT
C
C        TP=DRDT/V IS THE UNIT VECTOR IN THE INSTANTANEOUS VELOCITY DIRECTION
C
C        W=N CROSS WP  ,WHERE  WP IS THE SCAN VECTOR IN THE T,C PLANE
C
      CALL VECPRD(TP,T,DRPDOTT,1)
      CALL VECPRD(TP,SS,DRLDOTT,1)
      CALL VECPRD(CC,T,DRPDOTC,1)
      CALL VECPRD(CC,SS,DRLDOTC,1)
      CALL VECPRD(TP,W,TDOTW,1)
      CALL VECPRD(CC,W,CDOTW,1)
C
      DFDPHI=DRPDOTT*TDOTW + DRPDOTC*CDOTW
      DFDLAM=DRLDOTT*TDOTW + DRLDOTC*CDOTW
C        DF/D(PHI)=DFDPHI , DF/D(LAMDA)=DFDLAM
C
      DTDPHI=-DFDPHI/DFDT
      DTDLAM=-DFDLAM/DFDT
C

      DO 27 I=1,3
      DTPDPHI(I)=DTDT(I)*DTDPHI
```

```
   27 DTPDLAM(I)=DTDT(I)*DTDLAM
C
C         BELOW PHI,LAMDA COORESPOND TO THE GROUND TRACK COORDINATES
C
      C=COS(PHI)
      S=SIN(PHI)
      CL=COS(LAMDA)
      SL=SIN(LAMDA)
      RC=RX*C*CL + RY*C*SL + (A*A/B/B)*RZ*S
      DRCDPHI=-RX*S*CL - RY*S*SL + (A*A/B/B)*RZ*C
      DRCDLAM=-RX*C*SL + RY*C*CL
C
      TERM=DRCDPHI*DPHI + DRCDLAM*DLAMDA
      TERM2=DRDT(1)*C*CL + DRDT(2)*C*SL + A*A*DRDT(3)*S/B/B
C
      DRCDPHI=(TERM + TERM2)*DTDPHI
      DRCDLAM=(TERM + TERM2)*DTDLAM
C
      ANG=PIE/4. + ALPHAC/2.
      T1=TAN(ANG)
      T2=1./COS(ANG)/COS(ANG)
      TERM=WP(1)*TP(1) + WP(2)*TP(2) + WP(3)*TP(3)
      THETAS=-ASIN(TERM)
      D=RC*ALOG(T1)/COS(THETAS)
C
C         HERE WE BUILD THE PARTIALS OF THE C VECTOR.
C
      DO 85 I=1,3
      DRPHI(I)=T(I) - DRDT(I)*DTDPHI
   85 DRLAM(I)=SS(I) - DRDT(I)*DTDLAM
C
      TAU1=-S*CL*DPHI - C*SL*DLAMDA
      TAU2=-S*SL*DPHI + C*CL*DLAMDA
      TAU3=C*DPHI
C
      DNDPHI(1)=TAU1*DTDPHI
      DNDPHI(2)=TAU2*DTDPHI
      DNDPHI(3)=TAU3*DTDPHI
C
      DNDLAM(1)=TAU1*DTDLAM
      DNDLAM(2)=TAU2*DTDLAM
      DNDLAM(3)=TAU3*DTDLAM
C
      DVDPHI=DVDT*DTDPHI
      DVDLAM=DVDT*DTDLAM
      DO 5 I=1,3
    5 DUMMY(I)=DRDT(I)/V
      CALL CROSS(DNDPHI,DUMMY,E)
      DO 10 I=1,3
   10 DUMMY(I)=DDRDT(I)*DTDPHI/V
      CALL CROSS(N,DUMMY,F)
      DO 15 I=1,3
   15 DUMMY(I)=DRDT(I)*DVDPHI/V/V
      CALL CROSS(N,DUMMY,G)
C
C         BELOW WE COMPUTE THE DERIVATIVE OF THE C VECTOR W.R.T. PHI
C
```

```
      DO 20 I=1,3
   20 DCDPHI(I)=E(I) + F(I) - G(I)
C
      DO 25 I=1,3
   25 DUMMY(I)=DRDT(I)/V
      CALL CROSS(DNDLAM,DUMMY,E)
      DO 30 I=1,3
   30 DUMMY(I)=DDRDT(I)*DTDLAM/V
      CALL CROSS(N,DUMMY,F)
      DO 35 I=1,3
   35 DUMMY(I)=DRDT(I)*DVDLAM/V/V
      CALL CROSS(N,DUMMY,G)
C
C        BELOW WE COMPUTE THE DERIVATIVE OF THE C VECTOR W.R.T. LAMDA
C
      DO 40 I=1,3
   40 DCDLAM(I)=E(I) + F(I) - G(I)
      DO 22 I=1,3
   22 DWPDPHI(I)=DWPDT(I)*DTDPHI
      DO 42 I=1,3
   42 DWPDLAM(I)=DWPDT(I)*DTDLAM
C
C
      TERM1=DR(1)*CC(1) + DR(2)*CC(2) + DR(3)*CC(3)
      TERM2=DRPHI(1)*CC(1) + DRPHI(2)*CC(2) + DRPHI(3)*CC(3)
      TERM3=DR(1)*DCDPHI(1) + DR(2)*DCDPHI(2) + DR(3)*DCDPHI(3)
C
      DACDPHI=(RC*(TERM2 + TERM3) - TERM1*DRCDPHI)/(RC*RC + TERM1*TERM1)
C
      TERM2=DRLAM(1)*CC(1) + DRLAM(2)*CC(2) + DRLAM(3)*CC(3)
      TERM3=DR(1)*DCDLAM(1) + DR(2)*DCDLAM(2) + DR(3)*DCDLAM(3)
C
      DACDLAM=(RC*(TERM2 + TERM3) - TERM1*DRCDLAM)/(RC*RC + TERM1*TERM1)
C
      P=CONS(6)
      CALL FNTS2(TI,CONS,XCOEFFS,YCOEFFS,FCOEFFS,F,NX,NY,NF)
      CF=COS(F(4))
      SF=SIN(F(4))
      DDXDT=DVDT*CF - V*SF*DFT
      DDYDT=DVDT*SF + V*CF*DFT
C
      DDXDPHI=DDXDT*DTDPHI
      DDXDLAM=DDXDT*DTDLAM
      DDYDPHI=DDYDT*DTDPHI
      DDYDLAM=DDYDT*DTDLAM
C
      CALL VECPRD(DWPDPHI,TP,DWPDOTT,1)
      CALL VECPRD(DWPDLAM,TP,DWLDOTT,1)
      CALL VECPRD(WP,DTPDPHI,WDOTDTP,1)
      CALL VECPRD(WP,DTPDLAM,WDOTDTL,1)
C
      SUMSQ=SQRT(1. - TERM*TERM)
C
      DGDPHI=(DXDT*DDYDPHI - DYDT*DDXDPHI)/(DXDT*DXDT + DYDT*DYDT)
     1 -(DWPDOTT + WDOTDTP)/SUMSQ
C
      DGDLAM=(DXDT*DDYDLAM - DYDT*DDXDLAM)/(DXDT*DXDT + DYDT*DYDT)
```

```
     1 -(DWLDOTT + WDOTDTL)/SUMSQ
C
      CH=COS(THETAS)
      SH=SIN(THETAS)
C
      DDDPHI=(DRCDPHI*ALOG(T1) + RC*T2*DACDPHI/T1/2.)/CH
     1 - RC*ALOG(T1)*SH*(DWPDOTT + WDOTDTP)/SUMSQ/(CH*CH)
C
      DDDLAM=(DRCDLAM*ALOG(T1) + RC*T2*DACDLAM/T1/2.)/CH
     1 - RC*ALOG(T1)*SH*(DWLDOTT + WDOTDTL)/SUMSQ/(CH*CH)
      C=COS(F(3) + THETAS)
      S=SIN(F(3) + THETAS)
      DXDPHI=DXDT*DTDPHI
      DXDLAM=DXDT*DTDLAM
      DYDPHI=DYDT*DTDPHI
      DYDLAM=DYDT*DTDLAM
C        COMPUTE THE PARTIAL DERIVATIVES
C
C                  DX/D(PHI), DX/D(LAMDA), DY/D(PHI), DY/DLAMDA)
C
C        LET U = X
C            V = Y
C            1 = PHI
C            2 = LAMDA
C
      U1=DXDPHI + DDDPHI*C - D*S*DGDPHI
      U2=DXDLAM + DDDLAM*C - D*S*DGDLAM
      V1=DYDPHI + DDDPHI*S + D*C*DGDPHI
      V2=DYDLAM + DDDLAM*S + D*C*DGDLAM
      ESQ=2.*FF - FF*FF
      S=SIN(ANG2)
      W=SQRT(1. - ESQ*S*S)
      E1=U1*U1 + V1*V1
      E2=U2*U2 + V2*V2
C
C        COMPUTE LENGTH DISTORTIONS ALONG LINES OF CONST LAMDA
C
      HP=W*W*W*SQRT(E1)/( A*(1.-ESQ))
C
C        COMPUTE LENGTH DISTORTIONS ALONG LINES OF CONST PHI.
C
      K=W*SQRT(E2)/(A*COS(ANG2))
      RETURN
      END
```

```
      SUBROUTINE SETUP(XO,TO,T,CONS)
      DIMENSION XO(1),CONS(1)
      REAL MU
      REAL MUDAYS,MC
      REAL MU,N
      PIE=2.*ASIN(1.)
C        PERIOD IN SECONDS.
      P=(103.267)*60.0
      MU=398601.2
C
      RO=(P*SQRT(MU)/2./PIE)**(2./3.)
      VO=SQRT(MU/RO)
      RADDEG=1.745329252E-02
C        POSITION
      XO(1)=0.
      XO(2)=RO*SIN(9.*RADDEG)
      XO(3)=RO*COS(9.*RADDEG)
C        VELOCITY
      XO(4)=VO
      XO(5)=0.
      XO(6)=0.
C        TIME
      TO=0.
      CONS(21)=RO
C
C        ORBITAL ELEMENTS
C        ----------------
C
C        A      SEMI-MAJOR AXIS OF THE ORBIT
C        E      ECCENTRICITY OF THE ORBIT
C        ANGI   INCLINATION OF THE ORIBT
C        ANGOM  LONGITUDE OF THE ASCENDING NODE
C        ANGW   ARGUMENT OF PERIFOCUS
C        T      TIME OF PERIFOCAL PASSAGE
C
      TO=0.
      MU=3.986012E05
      SQMU=6.313487E02
C        POSITION
      X=XO(1)
      Y=XO(2)
      Z=XO(3)
C        VELOCITY
      DX=XO(4)
      DY=XO(5)
      DZ=XO(6)
C
      RO=SQRT(X*X + Y*Y + Z*Z)
      VO=SQRT(DX*DX + DY*DY + DZ*DZ)
      DO=X*DX + Y*DY + Z*DZ
      AI=2.0/RO - VO*VO/MU
C
C        SEMIMAJOR AXIS
C
      A=1.0/AI
      PERIOD=2.*PIE*(A**1.5)/SQRT(MU)
      N=SQMU/A/SQRT(A)
```

BEST AVAILABLE COPY

```
      T1=1.0 - RO/A
      T2=DO/SQMU/SQRT(A)
C
C        ECCENTRICITY
C
      E=SQRT(T1*T1 + T2*T2)
      EO=ATAN2(T2,T1)
      MO=EO - E*SIN(EO)
      TT=TO - MO/N
      DDO=MU*(1.0/RO - 1.0/A)
      IF(E .LE. 1.E-10)GO TO 10
      T=1.0/MU/E
      P1=T*(DDO*X - DO*DX)
      P2=T*(DDO*Y - DO*DY)
      P3=T*(DDO*Z - DO*DZ)
      P=A*(1.0 - E*E)
      HO=RO - P
      DHO=DO/RO
      T=1./SQMU/SQRT(P)
      Q1=T*(DHO*X - HO*DX)
      Q2=T*(DHO*Y - HO*DY)
      Q3=T*(DHO*Z - HO*DZ)
C
      GO TO 12
   10 CONTINUE
C
C        P AND Q VECTORS FOR THE CIRCULAR ORBIT SPECIAL CASE
C
      P1=X/RO
      P2=Y/RO
      P3=Z/RO
      Q1=DX/VO
      Q2=DY/VO
      Q3=DZ/VO
   12 CONTINUE
C
      W1=P2*Q3 - P3*Q2
      W2=P3*Q1 - P1*Q3
      W3=P1*Q2 - P2*Q1
C
      ANGI=ACOS(W3)
      ANGOM=ATAN2(W1,-W2)
      ANGW=ATAN2(P3,Q2)
      WRITE(6,100)
  100 FORMAT(1H1)
      WRITE(6,105)
  105 FORMAT(45X,*INPUT PARAMETERS DEFINING THE CURRENT RUN.*,///)
      WRITE(6,110)
  110 FORMAT(47X,* POSITION*,20X,* VELOCITY*//)
      WRITE(6,115)XO(1),XO(4),XO(2),XO(5),XO(3),XO(6)
  115 FORMAT(42X,F15.5,14X,F15.5)
      WRITE(6,120)
  120 FORMAT(///57X,* ORBITAL ELEMENTS*,//)
      WRITE(6,125)A
  125 FORMAT(38X,F15.5,5X,* SEMI-MAJOR AXIS OF THE ORBIT*)
      WRITE(6,130)E
  130 FORMAT(38X,F15.5,5X,* ECCENTRICITY OF THE ORBIT*)
```

```
      WRITE(6,135)ANGI
  135 FORMAT(38X,F15.5,5X,* INCLINATION OF THE ORBIT*)
      WRITE(6,140)ANGOM
  140 FORMAT(38X,F15.5,5X,* LONGITUDE OF THE ASCENDING NODE*)
      WRITE(6,145)ANGW
  145 FORMAT(38X,F15.5,5X,* ARGUMENT OF PERIFOCUS*)
      WRITE(6,150)TT
  150 FORMAT(38X,F15.5,5X,* TIME OF PERIFOCAL PASSAGE*)
      WRITE(6,155)PERIOD
  155 FORMAT(38X,F15.5,5X,* SATELLITE ORBITAL PERIOD*)
      RETURN
      END
```

```
      SUBROUTINE  INTEG(Z,CONS,TO,TF)
C
C******************************************************************************
C          THIS PROGRAM INTEGRATES TO DETERMINE THE FOURIER COEFFICIENTS FOR
C          THE SATELLITE GROUND TRACK COORDINATES AND F.
C******************************************************************************
C
      COMMON/COUNT/M
      COMMON/XYCOEFS/XCOEFFS,YCOEFFS,FCOEFFS,NX,NY,NF
      DIMENSION XCOEFFS(20),YCOEFFS(20),FCOEFFS(20)
      DIMENSION Z(1),CONS(1)
      DIMENSION F(4)
      ILAST=0
      PIE=2.*ASIN(1.)
      DELT=CONS(10)
      T=TO
C
      CALL ROC(0.,CONS)
C
      ILAST=0
      T=TO
      CALL ICS(Z,CONS,4)
      DO 5 II=1,500
      IF((T + DELT) .GE. TF) ILAST=1
      IF(ILAST.EQ.1) DELT=TF - T
C
      CALL RUNGE(T,4,DELT,CONS,Z)
C
      IF(ILAST.EQ.1) GO TO 50
    5 CONTINUE
   50 CONTINUE
C
C         COMPUTE DXDT AVERAGE
C
C         CONS(5) = PERIOD OF ORBIT
      CONS(30)=Z(1)/CONS(5)
      YAVG=ABS(Z(4))/CONS(5)
      DXAVG=CONS(30)
C
C
      DELT=CONS(10)
      ILAST=0
      T=TO
C
C         NTERMS = 3*(NU. OF COEFFICIENTS TO BE FOUND) + 3
C
      NTERMS=33
      CALL ICS(Z,CONS,NTERMS)
      Z(2)=YAVG
      CALL ROC(0.,CONS)
      DO 10 II=1,500
      IF((T + DELT) .GE. TF) ILAST=1
      IF(ILAST.EQ.1) DELT=TF - T
C
      CALL RUNGE(T,NTERMS,DELT,CONS,Z)
C
      IF(ILAST.EQ.1) GO TO 100
```

```
   10 CONTINUE
  100 CONTINUE
C         TERM IS THE SCALE FACTOR FOR THE FOURIER INTEGRALS
      TERM=2./CONS(5)
      DO 20 I=4,NTERMS
   20 Z(I)=Z(I)*TERM
      XCOEFFS(1)=DXAVG
      YCOEFFS(1)=YAVG
      NFIX=(NTERMS - 3)/3. + 1.E-03
      DO 30 I=1,NFIX
      XCOEFFS(I+1)=Z(I+3)
      YCOEFFS(I+1)=Z(I + 3 + NFIX)
   30 FCOEFFS(I)=Z(I + 3 + NFIX + NFIX)
      NX=NFIX
      NY=NFIX
      NF=NFIX
      WRITE(6,999)
  999 FORMAT(1H1)
      WRITE(6,1000)
 1000 FORMAT(42X,*  SATELLITE GROUND TRACK INTEGRATION*//)
      WRITE(6,1005)TO,T
 1005 FORMAT(48X,* INITIAL TIME=*,F10.4,/48X,* FINAL TIME=*,F12.4//)
      WRITE(6,1010)CONS(10)
 1010 FORMAT(32X, * INTEGRATION STEP SIZE = (SATELLITE PERIOD)/400. =*,
     1 F10.4,* SEC.*//)
      WRITE(6,1035)
 1035 FORMAT(23X,*   RESULTS OF GROUND TRACK INTEGRATION FOR THE TIME IN
     1CREMENT DELTA = (PERIOD)/100.*///)
      WRITE(6,1015)
 1015 FORMAT(21X,* XG*,14X,* YG*,16X,* F*, 17X,* PHI*,12X,*LAMDA*,12X,
     1        * TIME*/)
      DT=CONS(5)/100.
      T=-DT
      DO 500 I=1,101
      T=T + DT
      CALL ROC(T,CONS)
      CALL FNTS2(T,CONS,XCOEFFS,YCOEFFS,FCOEFFS,F,NX,NY,NF)
      WRITE(6,1030)F(1),F(2),F(4),CONS(14),CONS(15),T
 1030 FORMAT(12X,5E18.7,F12.4)
  500 CONTINUE
      RETURN
      END
```

BEST AVAILABLE COPY

```
      SUBROUTINE ANG(C,S,T,N)
C
C***************************************************************************
C         THIS PROGRAM COMPUTES MULTIPLE ANGLE SINE AND COSINE TERMS FOR THE
C         ARGUMENT T = (PIE)*(TIME)/(PERIOD).
C
C         TIME = CURRENT TIME IN THE PROGRAM
C         PERIOD = SATELLITE ORBITAL PERIOD
C***************************************************************************
C
      DIMENSION C(1),S(1)
      C(1)=COS(T)
      S(1)=SIN(T)
      C(2)=C(1)*C(1) - S(1)*S(1)
      S(2)=2.*S(1)*C(1)
      IF(N.LE.2)RETURN
      DO 1 I=3,N
      S(I)=2.*S(I - 1)*C(1) - S(I - 2)
    1 C(I)=2.*C(I - 1)*C(1) - C(I - 2)
      RETURN
      END
```

```
      SUBROUTINE FNTS2(T,CONS,XCOEFFS,YCOEFFS,FCOEFFS,F,NX,NY,NF)
C
C***************************************************************************
C        THIS PROGRAM COMPUTES THE SATELLITE MAP PLANE X,Y COORDINATES
C        AND THE F-FUNCTION .  INADDITION THE 1-ST DERVATIVES OF X,Y, AND F
C        ARE COMPUTED.
C
C        A CALL TO ROC FOR THE TIME T MUST PRECEED A CALL TO THIS PROGRAM
C        CONS(18) = V(T) AT TIME T IN THE MAP PLANE
C
C        INPUT
C            T          = CURRENT TIME IN PROGRAM
C            XCOEFFS,NX= X-FOURIER FIT COEFFICIENTS AND NUMBER OF COEFFICIENTS
C            YCOEFFS,NY= Y-FOURIER FIT COEFFICIENTS AND NUMBER OF COEFFICIENTS
C            FCOEFFS,NF= F-FOURIER FIT COEFFICIENTS AND NUMBER OF COEFFICIENTS
C
C        OUTPUT
C            F(1)      = X GROUND TRACK
C            F(2)      = Y GROUND TRACK
C            F(3)      = TC-FRAME ROTATION ANGLE GAMMA
C            F(4)      = F-INTEGRAL OF V(T)/(RADIUS OF CURVATURE)
C            DX        = DXG/DT         , WHERE XG = X-GROUND TRACK COORDINATE
C            DY        = DYG/DT         , WHERE YG = Y-GROUND TRACK COORDINATE
C            DFT       = DF/DT
C***************************************************************************
C
      DIMENSION XCOEFFS(1),YCOEFFS(1),FCOEFFS(1),F(1),CONS(1)
      DIMENSION C(20),S(20)
      COMMON/DF/DFT
      COMMON/DXYMAP/DX,DY
      PIE=2.*ASIN(1.)
      CALL SETCON(A,B,PIE,PERIOD,CONS)
      NFIX=NX
      IF((NY/NX).GT.1)NFIX=NY
C        TIME MUST BE NORMALIZED TO T=PIE*T/PERIOD FOR ANG SUB CALL
      TP=PIE*T/PERIOD
C
      CALL ANG(C,S,TP,NFIX)
C        COMPUTE Y-GROUND TRACK COORDINATE
      Y=0.0
      DO 1 I=2,NY
    1 Y=Y + YCOEFFS(I)*C(I - 1)
C        COMPUTE X-GROUND TRACK COORDINATE
      X=XCOEFFS(1)*T
      DO 2 I=2,NX
    2 X=X + XCOEFFS(I)*S(I-1)
      FT=0.
      DO 6 I=1,NF
    6 FT=FT + FCOEFFS(I)*S(I)
      F(1)=X
      F(2)=Y
      F(4)=FT
      TERM=PIE/PERIOD
C        COMPUTE DX/DT FOR GROUND TRACK
      DX=CONS(18)*COS(F(4))
C        COMPUTE DY/DT FOR GROUND TRACK
      DY=CONS(18)*SIN(F(4))

      DF=0.
      DO 8 I=1,NF
    8 DF=DF + I*FCOEFFS(I)*C(I)
      DFT=DF*TERM
      F(3)=PIE/2. + ATAN2(DY,DX)
      RETURN
      END
```

```
      SUBROUTINE SETCON(A,B,PIE,PERIOD,CONS)
C
C*****************************************************************************
C         THIS PROGRAM SETS VALUES OF PARAMETERS USED IN THE CALCULATION OF
C         THE SATELLITE GROUND TRACK
C
C         NSTEPS = NUMBER OF STEPS IN THE INTEGRATION OF S.G.T.
C         DELT = TIME INCREMENT IN EACH INTEGRATION STEP
C*****************************************************************************
C
      DIMENSION CONS(1)
      F=1./298.3
      ESQ=2.*F - F*F
      A=6378.165
      B=A*SQRT(1. - ESQ)
      PIE=2.*ASIN(1.)
      PERIOD=103.267*60.
      NSTEPS=400
      DELT=PERIOD/NSTEPS
      CONS(6)=PERIOD
      CONS(7)=F
      CONS(8)=A
      CONS(9)=B
      CONS(10)=DELT
      CONS(13)=0.
C
      RETURN
      END
```

```
      SUBROUTINE INVERSE(TO,TI,CONS,FF,X,Y,PHI,LAMDA)
C
C*****************************************************************************
C         THIS PROGRAM COMPUTES PHI AND LAMDA GIVEN X AND Y IN THE MAP PLANE.
C
C         INPUT
C              X           = ABSCISSA
C              Y           = ORDINATE
C
C         OUTPUT
C              PHI         = GEODETIC LATITUDE
C              LAMDA       = LONGITUDE
C              TI          = TIME DISPLACEMENT VECTOR DELTA R LIES ALONG THE SCAN
C                              VECTOR W.
C*****************************************************************************
C
      COMMON/MSDATA/DDRDT,DRDT,C,H,XX,N
      COMMON/SCAN/TF,DFDT,W,DWDT,DT,T
      COMMON/DE/DET
      COMMON/DXYMAP/DX,DY
      COMMON/XYCOEFFS/XCOEFFS,YCOEFFS,FCOEFFS,NX,NY,NF
      COMMON/INERTAL/XN,DXN,DDXN
      COMMON/ANG/ALPHAC
      DIMENSION DC(3),DN(3),DWDT(3),CONS(1),ROT(3,3),XX(6),N(3)
      DIMENSION U(3),DU(3),XN(3),DXN(3),DDXN(3),W(3),DW(3),T(3),DT(3)
      DIMENSION F(4),XCOEFFS(20),YCOEFFS(20),FCOEFFS(20)
      DIMENSION C(3),DDRDT(3),DRDT(3),E(3)
      DIMENSION TP(3)
      COMMON/TIME/TIME2,J
      REAL M,LAMDA,N
      REAL LAMDAO
      TOLD=1.E99
C
      CALL SETCON(A,B,PIE,P,CONS)
      CALL ROC(P,,CONS)
C
C          GET THE INITIAL GUESS ON TIME
C
C          XCOEFFS(1) IS APPROXIMATELY THE AVERAGE VELOCITY.
      TI=X/XCOEFFS(1)
C
      LAST=0
      ICOUNT=0
      DO 1000 II=1,15
    8 CALL ROC(TI,CONS)
      CALL ROTATE(ROT,TO,TI,WE)
      V=CONS(19)
      DVDT=CONS(17)
C
C          XN = POSITION IN INERTIAL SPACE
C         DXN = VELOCITY IN INERTIAL SPACE
C        DDXN = ACCELERATION IN INERTIAL SPACE
C
C          U IS THE VECTOR IN THE SCAN DIRECTION
C
      CALL CROSS(XN,DXN,U)
C
```

BEST AVAILABLE COPY

```
C
      CALL CROSS(XN,DDXN,DU)
C
C
C         DU IS THE DERIVATIVE OF THE SCAN VECTOR IN THE E-FRAME WITH COMPONENTS
C         IN INERITAL SPACE
C
      DU(1)=DU(1) + WE*U(2)
      DU(2)=DU(2)  - WE*U(1)
      DU(3)=DU(3)
C
C
C         ROTATE THE COMPONENTS OF U,DU TO THE E-FRAME.
C
      CALL EFRAME(U,ROT,3)
      CALL EFRAME(DU,ROT,3)
C
C         T IS THE UNIT VECTOR ALONG TRACK  WITH COMPONENTS IN THE E-FRAME
      DO 15 I=1,3
      T(I)=DRDT(I)/V
   15 DT(I)=DDRDT(I)/V - DVDT*DRDT(I)/(V*V)
C
C         C IS THE UNIT VECTOR IN THE CROSS TRACK DIRECTION  WITH COMPONENTS IN
C         THE E-FRAME.
C
      CALL CROSS(U,T,C)
      DPHI=CONS(26)
      DLAMDA=CONS(27)
      CGT=COS(CONS(14))
      SGT=SIN(CONS(14))
      CLGT=COS(CONS(15))
      SLGT=SIN(CONS(15))
C
C         DN IS THE DERIVATIVE OF THE NORMAL VECTOR IN THE E-FRAME
C
      DN(1)=-SGT*CLGT*DPHI - CGT*SLGT*DLAMDA
      DN(2)=-SGT*SLGT*DPHI + CGT*CLGT*DLAMDA
      DN(3)=CGT*DPHI
C
      CALL CROSS(DN,T,E)
      CALL CROSS(N,DT,F)
C
C         DC IS THE DERIVATIVE OF THE C VECTOR IN THE E-FRAME.
C
      DO 19 I=1,3
   19 DC(I)=E(I) + F(I)
C

      T1=DU(1)*T(1) + DU(2)*T(2) + DU(3)*T(3)
      T2=U(1)*DT(1) + U(2)*DT(2) + U(3)*DT(3)
      T3=U(1)*T(1) + U(2)*T(2) + U(3)*T(3)
      T4=DU(I)*C(1) + DU(2)*C(2) + DU(3)*C(3)
      T5=U(1)*DC(1) + U(2)*DC(2) + U(3)*DC(3)
      T6=U(1)*C(1) + U(2)*C(2) + U(3)*C(3)
C
      TERM=T3*(T1+T2) + T6*(T4+T5)
      SUMSQ=SQRT(T3*T3 + T6*T6)
```

BEST AVAILABLE COPY

```
      AA=T3/SUMSQ
      BB=T6/SUMSQ
C
C         W IS THE SCAN VECTOR AS PROJECTED ONTO THE T,C PLANE
C
      DO 45 I=1,3
   45 W(I)=AA*T(I) + BB*C(I)
C
      DA=(T1+T2)/SUMSQ - T3*TERM/(SUMSQ*SUMSQ*SUMSQ)
      DB=(T4+T5)/SUMSQ - T6*TERM/(SUMSQ*SUMSQ*SUMSQ)
C
C        DWDT IS THE DERIVATIVE OF THE SCAN VECTOR W IN THE TC-FRAME.
C
      DO 51 I=1,3
   51 DWDT(I)=DA*T(I) + AA*DT(I) + DB*C(I) + BB*DC(I)
C
      CALL CROSS(N,W,TP)
      CALL FNTS2(TI,CONS,XCOEFFS,YCOEFFS,FCOEFFS,F,NX,NY,NF)
      THETA=F(3)-PIE/2.
      CT=COS(THETA)
      ST=SIN(THETA)
      DDX=DVDT*COS(F(4)) - V*SIN(F(4))*DFT
      DDY=DVDT*SIN(F(4)) + V*COS(F(4))*DFT
      DTHETA=(DX*DDY - DY*DDX)/(DX*DX + DY*DY)
      A1=AA
      A2=BB
      DA1=DA
      DA2=DB
      B1=-(A1*ST + A2*CT)
      B2=A1*CT - A2*ST
C
C        FT = (DELTA-R).(N-CROSS-W)  , FT = 0.0 WHEN TIME = T-STAR.
C
      FT=B1*(X - F(1)) + B2*(Y - F(2))
C
      DB1=-((DA1 - DTHETA*A2)*ST + (DA2 + DTHETA*A1)*CT)
      DB2=(DA1 - DTHETA*A2)*CT - (DA2 + DTHETA*A1)*ST
C
      DFDT=-B1*DX - B2*DY + DB1*(X - F(1)) + DB2*(Y - F(2))
C
      IF(LAST.EQ.1000)GO TO 200
C
C        FIND NEW TIME ESTIMATE VIA. NEWTONS METHOD
C
      TI=TI - FT/DFDT
C
      IF(ABS(TOLD-TI).LT.1.E-06) LAST=1000
      TOLD=TI
      DT=TOLD-TI
 1000 CONTINUE
      WRITE(6,352)
  352 FORMAT(1H1,//* FAILED TO CONVERGE IN INVERSE T-STAR SEARCH*///)
      CALL EXIT
  200 CONTINUE
      TIME2=TI
C
C        USE PHI,LAMDA OF THE GROUND TRACK AS STARTING GUESSES FOR 2-D NEWTONS
```

BEST AVAILABLE COPY

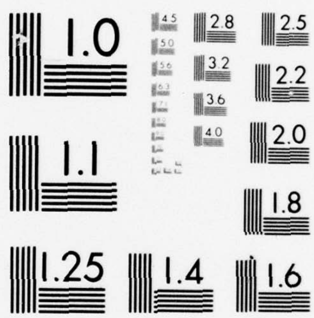
```
C         METHOD FOR THE PHI, LAMDA OF POINT OFF GROUND TRACK.
C
      PHIO=CONS(14)
      LAMDAO=CONS(15)
      DO 60 I=1,10
      CS=COS(PHIO)
      SS=SIN(PHIO)
      CSL=COS(LAMDAO)
      SSL=SIN(LAMDAO)
      RADIUS=A*A/SQRT(A*A*CS*CS + B*B*SS*SS)
C
C         CURRENT EARTH FIXED COORDINATE ESTIMATES
C
      R1=RADIUS*CS*CSL
      R2=RADIUS*CS*SSL
      R3=B*B*RADIUS*SS/A/A
C
      DR1=R1 - (XX(1) - H*N(1))
      DR2=R2 - (XX(2) - H*N(2))
      DR3=R3 - (XX(3) - H*N(3))
      TI=TIME2
C
C         ADJUST TIME FOR SCAN RATE
C
      CALL SCANNER(TI,R1,R2,R3,XX(1),XX(2),XX(3),N(1),N(2),N(3),H,
     1             DR1,DR2,DR3,C(1),C(2),C(3))
C
C         GET NEW MAP PLANE X,Y ESTIMATES
C
      CALL US(CONS,PHIO,LAMDAO,0.,X0,Y0,100)
C
C
C         GET PARTIAL DERIVATIVES
C
      CALL ERRORS(A,B,CONS,FF,HP,K,U1,U2,V1,V2,LAMDAO,PHIO,TI)
      DELT=1./(U1*V2-U2*V1)
C
C         COMPUTE NEW PHI, LAMDA ESTIMATES.
C
      PHI=PHIO + DELT*((X-X0)*V2 - (Y-Y0)*U2)
      LAMDA=LAMDAO + DELT*(-(X-X0)*V1 + (Y-Y0)*U1)
      IF(ABS(PHI-PHIO).LE.1.E-08.AND.ABS(LAMDA-LAMDAO).LE.1.E-08)GOTO70
      PHIO=PHI
      LAMDAO=LAMDA
   60 CONTINUE
      WRITE(6,1500)
 1500 FORMAT(1H1,* FAILED TO CONVERGE IN INVERSE PHI,LAMDA SEARCH*//)
      CALL EXIT
   70 CONTINUE
      RETURN
      END
```

BEST AVAILABLE COPY

AD-A052 143 VIRGINIA UNIV CHARLOTTESVILLE RESEARCH LABS FOR THE--ETC F/G 8/2
FORMULATION OF A SPACE OBLIQUE MERCATOR MAP PROJECTION.(U)
FEB 78 J L JUNKINS, J D TURNER DAAG53-76-C-0067
UNCLASSIFIED UVA/525023/ESS77/105 ETL-0131 NL

2 OF 2
AD
A052143

END
DATE
FILMED
5-78
DDC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

```
      SUBROUTINE RUNGE(TK,N,DELT,CONST,XK)
C
C*******************************************************************************
C         THIS PROGRAM USES A 4-CYCLE RUNGA-KUTTA ALGORITHM TO NUMERICALLY
C         INTEGRATE THE STATE VECTOR (XK) FROM TIME TK TO (TK + DELT).
C*******************************************************************************
C
      DIMENSION X(40),XK( 1),F(40),D1(40),D2(40),D3(40),D4(40),CONST( 1)
C
      T=TK
      DO 10 I=1,N
   10 X(I)=XK(I)
      CALL ROC(T,CONST)
      CALL DERIV(T,N,X,CONST,F)
C
      DO 20 I=1,N
   20 D1(I)=DELT*F(I)
C
      T=TK+DELT/2.0
      DO 30 I=1,N
   30 X(I)=XK(I)+D1(I)/2.0
      CALL ROC(T,CONST)
      CALL DERIV(T,N,X,CONST,F)
C
      DO 40 I=1,N
   40 D2(I)=DELT*F(I)
C
      DO 50 I=1,N
   50 X(I)=XK(I)+D2(I)/2.0
      CALL DERIV(T,N,X,CONST,F)
C
      DO 60 I=1,N
   60 D3(I)=DELT*F(I)
C
      T=TK+DELT
      DO 70 I=1,N
   70 X(I)=XK(I)+D3(I)
      CALL ROC(T,CONST)
      CALL DERIV(T,N,X,CONST,F)
C
      DO 80 I=1,N
   80 D4(I)=DELT*F(I)
C
C
      TK=T
      DO 90 I=1,N
   90 XK(I)=XK(I)+(D1(I)+2.0*D2(I)+2.0*D3(I)+D4(I))/6.0
C
      RETURN
      END
```

```
      SUBROUTINE ICS(Z,CONS,NFIX)
C
C*****************************************************************************
C         THIS PROGRAM SETS THE INITIAL CONDITIONS FOR THE INTEGRATION OF THE
C         SATELLITE GROUND TRACK AND THE FOURIER SERIES COEFFICIENTS
C*****************************************************************************
C
      DIMENSION Z(1),CONS(1)
      DIMENSION XO(6)
      COMMON/ROCDATA/A,B,XO,TO
C
      PIE=2.*ASIN(1.)
      DO 5 I=1,NFIX
    5 Z(I)=0.
C
      RETURN
      END
```

```
      SUBROUTINE DERIV(T,N,X,CONST,F)
C
C*******************************************************************************
C         THIS PROGRAM COMPUTES THE FUNCTION APPROXIMATIONS REQUIRED
C         IN SUBROUTINE RUNGE.
C*******************************************************************************
C
      DIMENSION X( 1),F(40),CONST( 1)
      DIMENSION C(20),S(20)
C         CONST(17)=DOSDT
C         CONST(18)=DSDT
C         CONST(28)=1./ROC   , WHERE ROC= RADIUS OF CURVATUVE
      P=CONST(6)
      PIE=2.*ASIN(1.)
      F(1)=CONST(18)*COS(X(3))
      F(2)=CONST(18)*SIN(X(3))
      F(3)=CONST(18)*CONST(28)
C         IF N=4 WE INTEGRATE TO FIND DXAVG AND YAVG
C
      IF(N.EQ.4) F(4)=X(2)
      IF(N.EQ.4) GO TO 100
      DXAVG=CONST(30)
      XTERM=X(1) - DXAVG*T
      YTERM=X(2)
      FTERM=X(3)
C         NORMALIZE TIME
      TP=PIE*T/P
C
C         NTERMS ALLOWS US TO VARY THE NUMBER OF COEFFICIENTS TO BE FIT
C
      NTERMS=(N - 3)/3. + 1.E-03
      NFIX=NTERMS + 3
      CALL ANG(C,S,TP,NTERMS)
      DO 5 I=4,NFIX
      F(I)=XTERM*S(I - 3)
      F(I + NTERMS)=YTERM*C(I - 3)
    5 F(I + NTERMS + NTERMS)=FTERM*S(I -3)
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ROC(T,CONS)
C
C***************************************************************************
C        THIS PROGRAM COMPUTES THE RADIUS OF CURVATURE AND THE FIRST AND
C        SECOND DERIVATIVES OF THE MOTION ALONG THE SATELLITE GROUND TRACK.
C***************************************************************************
C
      DIMENSION CC(3),N(3),DDRDT(3),DDRDS(3),DRDS(3),DRDT(3)
      DIMENSION V(3),D(3),W(3),M(3,3),WV(3),WWR(3),DHDT(3),DDHDT(3)
      DIMENSION XO(6),X(6),CONS(1)
      DIMENSION XN(3),DXN(3),DDXN(3)
      COMMON/ROCDATA/A,B,XO,TO
      COMMON/NSDATA/DDRDT,DRDT,CC,H,X,N
      COMMON/INERTAL/XN,DXN,DDXN
      REAL LAMDA,M,MU,N
      MU=398601.2
C
C
C
      CALL ORBIT(XO,X,TO,T,CONS)
C
C        XN   = IS THE SATELLITE POSITION IN INERTIAL SPACE
C        DXN  = IS THE SATELLITE VELOCITY IN INERTIAL SPACE
      DO 13 I=1,3
      XN(I)=X(I)
   13 DXN(I)=X(I+3)
      DO 10 I=1,3
   10 V(I)=X(I+3)
C
      CALL ROTATE(M,TO,T,RATE)
C
C
C        R=R(T), T=TIME ,RSQ=R(T)*R(T)
C
      RSQ=X(1)*X(1) + X(2)*X(2) + X(3)*X(3)
      R=SQRT(RSQ)
      R3=RSQ*SQRT(RSQ)
C
C        W      IS THE ANGULAR VELOCITY OF THE EARTH W.R.T. INERTIAL SPACE
C
      W(1)=0.
      W(2)=0.
      W(3)=RATE
C
C
      CALL CROSS(W,X,D)
C
C        COMPUTE THE E-FRAME VELOCITY WITH COMPONENTS IN INERTIAL SPACE
C
      DO 8 I=1,3
    8 DRDT(I)=V(I) - D(I)
C
C        COMPUTE THE E-FRAME VELOCITY
C
      CALL EFRAME(DRDT,M,3)
C
C        WV       IS THE CORIOLIS ACCELERATION
```

BEST AVAILABLE COPY

```
C
C
      CALL CROSS(W,V,WV)
C
C        WWR      IS THE CENTRIPETAL ACCELERATION
C
      CALL CROSS(W,C,WWR)
C
C        DDRDT   IS THE ACCELERATION OF THE SATELLITE IN E-FRAME WITH
C        COMPONENTS IN INERTIAL SPACE
C
      DO 11 I=1,3
   11 DDRDT(I)=-MU*X(I)/R3 - 2.*WV(I) + WWR(I)
C        DDXN = IS THE SATELLITE ACCELERATION IN INERTIAL SPACE
      DO 15 I=1,3
   15 DDXN(I)=-MU*XN(I)/R3
C
C        COMPUTE THE E-FRAME COMPONENTS OF THE ACCELERATION
C
      CALL EFRAME(DDRDT,M,3)
C
C        COMPUTE THE E-FRAME COMPONENTS OF THE POSITION VECTOR
C
      CALL EFRAME(X,M,3)
C
      DOWN=X(1)*X(1) + X(2)*X(2)
      LAMDA=ATAN2(X(2),X(1))
      DLAMDA=(X(1)*DRDT(2) - X(2)*DRDT(1))/DOWN
      DDLAMDA=(X(1)*DDRDT(2) - X(2)*DDRDT(1))/DOWN - 2.*DLAMDA*
     1   (X(1)*DRDT(1) + X(2)*DRDT(2))/DOWN
C
      RXY=SQRT(X(1)*X(1) + X(2)*X(2))
C
      CALL PHIH(RXY,X(3),R,PHI,H,LAMDA,A,B)
C
      CALL DPHIDH(A,B,H,HP,HPP,PHI,DPHI,DDPHI,X,RXY,R,MU,DRDT,DDRDT)
C
      C=COS(PHI)
      S=SIN(PHI)
      CL=COS(LAMDA)
      SL=SIN(LAMDA)
C
C        COMPUTE E-COMPONENTS OF DH/DT
C
      DHDT(1)=HP*C*CL - H*(S*CL*DPHI + C*SL*DLAMDA)
      DHDT(2)=HP*C*SL - H*(S*SL*DPHI - C*CL*DLAMDA)
      DHDT(3)=HP*S + H*C*DPHI
C
C        COMPUTE E-COMPONENTS OF D2H/DT2
C
      DDHDT(1)=HPP*C*CL - 2.*HP*(S*CL*DPHI + C*SL*DLAMDA)
     1          +H*(-C*CL*(DPHI*DPHI + DLAMDA*DLAMDA) + 2.*S*SL*DPHI
     2          *DLAMDA - S*CL*DDPHI - C*SL*DDLAMDA)
      DDHDT(2)=HPP*C*SL - 2.*HP*(S*SL*DPHI - C*CL*DLAMDA)
     1         +H*(-C*SL*(DPHI*DPHI + DLAMDA*DLAMDA) -2.*S*CL*DPHI
     2         *DLAMDA + C*CL*DDLAMDA -S*SL*DDPHI)
      DDHDT(3)=HPP*S + 2.*HP*C*DPHI + H*(C*DDPHI - S*DPHI*DPHI)
```

BEST AVAILABLE COPY

```
C
C        DRDT    IS THE VELOCITY OF THE SATELLITE G.T. IN THE E-FRAME.
      DO 9 I=1,3
    9 DRDT(I)=DRDT(I) - DHDT(I)
C
C        COMPUTE THE MAGNITUDE OF THE VELOCITY ALONG THE SPACE CURVE
C
      SUM=0.
      DO 12 I=1,3
   12 SUM=SUM + DRDT(I)*DRDT(I)
      DSDT=SQRT(SUM)
C
C        DDRDT   IS THE ACCELERATION OF THE SATELLITE G.T. IN THE E-FRAME.
C
      DO 16 I=1,3
   16 DDRDT(I)=DDRDT(I) - DDHDT(I)
C
C        COMPUTE THE COMPONENTS OF THE UNIT VECTOR ALONG THE SPACE CURVE
C
      DO 17 I=1,3
   17 DRDS(I)=DRDT(I)/DSDT
C
C        CONST      IS THE DOT PRODUCT OF THE VELOCITY AND ACCELERATION
C
      CONST=DDRDT(1)*DRDT(1) + DDRDT(2)*DRDT(2) + DDRDT(3)*DRDT(3)
      VSQ=DSDT*DSDT
C
C        N(I),I=1,3 ARE THE COMPONENTS OF THE UNIT VECTOR FROM THE
C        THE SATELLITE GROUND TRACK TO THE SATELLITE.
C
      N(1)=C*CL
      N(2)=C*SL
      N(3)=S
C
      CALL CROSS(N,DRDS,CC)
C
C        ROSQ=1./(RCC)**2.
C
C        HERE WE MAKE USE OF THE FACT THE CC IS PERPENDICULAR TO DRDT
C
      TERM=(DDRDT(1)*CC(1) + DDRDT(2)*CC(2) + DDRDT(3)*CC(3))/VSQ
      ROSQ=TERM*TERM
C
      SUM=0.
      DO 19 I=1,3
   19 SUM=SUM + DRDT(I)*DDRDT(I)
      DDSDT=SUM/DSDT
C
      CONS(14)=PHI
      CONS(15)=LAMDA
      CONS(17)=DDSDT
      CONS(18)=DSDT
      CONS(19)=ROSQ
      CONS(26)=DPHI
      CONS(27)=DLAMDA
      CONS(28)=TERM
C
      RETURN
   50 FORMAT(//10X,I5)
      END
```

```
      SUBROUTINE ROTATE(M,TO,T,WE)
      DIMENSION M(3,3)
      REAL M
C
C        ANGULAR RATE AS RADIANS PER MEAN SIDERIAL DAY.
C
      PIE=3.14159265
      WE=2.*PIE/86164.09054
      THETAO=0.
      THETA=THETAO + WE*(T - TO)
      C=COS(THETA)
      S=SIN(THETA)
      DO 1 I=1,3
      DO 1 J=1,3
    1 M(I,J)=0.
      M(1,1)=C
      M(1,2)=S
      M(2,1)=-S
      M(2,2)=C
      M(3,3)=1.
      RETURN
      END
```

BEST AVAILABLE COPY

```
      SUBROUTINE PHIH(RXY,Z,RN,PHI,H,LAMDA,A,B)
C
C        THIS PROGRAM COMPUTES PHI AND H BY A 2-DIMENSIONAL TAYLOR SERIES.
C
C        RXY = (X*X + Y*Y)**0.5
C
      R=A
      PHI=ASIN(Z/RN)
      H=RN - R
C
      DO 1 I=1,10
      C=COS(PHI)
      S=SIN(PHI)
      TERM=A*A*C*C + B*B*S*S
      TERM2=A*A/SQRT(TERM)
      Q=A*A*(A*A -B*B)*SIN(2.*PHI)/2./(TERM**1.5)
      E11=Q*C - (TERM2 + H)*S
      E12=C
      E21=B*B*(Q*S + TERM2*C)/A/A + H*C
      E22=S
      DELT=E11*E22 - E12*E21
      DIFF1=RXY - (TERM2 + H)*C
      DIFF2=Z  - (B*B*TERM2/A/A + H)*S
      PHIN=PHI + (DIFF1*E22 - DIFF2*E12)/DELT
      HN=H + (-DIFF1*E21 + DIFF2*E11)/DELT
      IF(ABS(PHIN-PHI).LE.1.E-06.AND.ABS(HN-H).LE.1.E-06)GO TO 10
      PHI=PHIN
    1 H=HN
      WRITE(6,5)
    5 FORMAT(*1H1*,//*  FAILED TO CONVERGE IN PHIH*)
      STOP
   10 PHI=PHIN
      H=HN
      RETURN
      END
```

BEST AVAILABLE COPY

```
      SUBROUTINE CROSS(A,B,C)
      DIMENSION A(1),B(1),C(1)
C
C        THIS PROGRAM COMPUTES THE CROSS PRODUCT OF TWO VECTORS.
C
      C(1)=A(2)*B(3) - A(3)*B(2)
      C(2)=A(3)*B(1) - A(1)*B(3)
      C(3)=A(1)*B(2) - A(2)*B(1)
      RETURN
      END
```

```
      SUBROUTINE EFRAME(A,B,N)
C
C        THIS PROGRAM ROTATES COORDINATES BETWEEN DIFFERENT FRAMES
C
      DIMENSION A(1),B(3,3),C(3)
      DO 1 I=1,N
    1 C(I)=A(I)
      DO 2 I=1,N
    2 A(I)=B(I,1)*C(1) + B(I,2)*C(2)  + B(I,3)*C(3)
      RETURN
      END
```

```
      SUBROUTINE ORBIT (X,XF,TI,TF,CONS)
C
C***********************************************************************
C
C     THIS PROGRAM DETERMINES THE STATE HISTORY VIA THE CHANGE IN
C     ECCENTRIC ANOMALY SOLUTION
C
C***********************************************************************
C
C
C     X(I)    IS THE INITIAL STATE VECTOR
C     XF(I)   IS THE FINAL STATE VECTOR
C     RO      IS THE INITIAL RADIUS
C     VO      IS THE INITIAL VELOCITY
C     RN      IS THE CURRENT RADIUS
C     A       IS THE SEMI-MAJOR AXIS
C     D       IS THE DOT PRODUCT OF POSITION AND VELOCITY(INITIAL)
C
C
C***********************************************************************
C
      DIMENSION X( 1),XF( 1)
      DIMENSION CONS( 1)
      REAL MU
      CONS(20)=2.*ASIN(1.)
      CONS(1)    =398601.2
      MU=CONS(1)
      PIE=CONS(20)
C     INITIAL STATE    (INPUT)
C
      RO=X(1)*X(1) + X(2)*X(2) + X(3)*X(3)
      VO=X(4)*X(4) + X(5)*X(5) + X(6)*X(6)
C
      RO=SQRT(RO)
      VO=SQRT(VO)
      AP=2.0/RO - VO*VO/MU
      IF(AP .LE. 0.0) GO TO 500
      A=1.0/AP
      PPP=(2.0*PIE*A*SQRT(A))/SQRT(MU)
C
      D=X(1)*X(4) + X(2)*X(5) + X(3)*X(6)
C
      CONS(2)       =1.0 - RO/A
      CONS(3)   =D/SQRT(MU*A)
      CONS(4)   =SQRT(MU)/(A*SQRT(A))
      CONS(5)=PPP
      YY=CONS(4)
      T=TF
      DELT=T -TI
      DELT=AMOD(DELT,PPP)
    5 T=DELT
C
      CALL NEWTON(CONS,PHI,TI,TF)
C
      S=SIN(PHI)
      C=COS(PHI)
      F=1.0 - (A/RO)*(1.0 - C)
```

```
      G=(T-TI)-(1.0/YY)*(PHI-S)
      SUM=0.0
C        POSITION AT TIME TF
      DO 140 I=1,3
      IP3=I+3
      XF(I)=F*X(I) + G*X(IP3)
  140 SUM=XF(I)*XF(I) + SUM
      RN=SQRT(SUM)
      FD=-(SQRT(MU*A)/(RO*RN))*S
      GD=1.0-(A/RN)*(1.0-C)
C        VELOCITY AT TIME TF
      DO 145 I=1,3
      IP3=I+3
  145 XF(IP3)=FD*X(I) + GD*X(IP3)
C
      GO TO 1000
  500 WRITE(6,300)
  300 FORMAT(1H1)
      WRITE(6,250)
  250 FORMAT(///52H  ERROR:  HYPERBOLIC EXCESS SPEED ACHIEVED IN ORBIT
     1     )
      STOP
 1000 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE NEWTON(CONS,PHI,TI,TF)
C
C**************************************************************************
C        THIS PROGRAM USES NEWTONS METHOD TO ITERATIVELY SOLVE FOR THE ANGLE
C        PHI THAT APPEARS IN KEPLERS EQUATION.
C**************************************************************************
C
      DIMENSION CONS( 1)
      YPI=CONS(4)*(TF-TI)
      DEL=AMOD(YPI,2.*CONS(20))
C     STARTING VALUE FOR NEWTONS METHOD
      PHI=DEL
      C=COS(PHI)
      S=SIN(PHI)
      EPSILON=1.E-12
      DO 100 I=1,10
      W=CONS(2)
      Z=CONS(3)
      Y=CONS(4)
      PHIN=PHI-(PHI-W*S+Z*(1.0-C)-    DEL )/(1.0-W*C+Z*S)
      IF(ABS(PHIN-PHI) .LT. EPSILON) GO TO 101
      PHI=PHIN
      C=COS(PHI)
      S=SIN(PHI)
  100 CONTINUE
      WRITE(6,10)
   10 FORMAT(1H1,*    FAILED TO CONVERGE IN SUBROUTINE NEWTON*//)
      STOP
  101 CONTINUE
      PHI=PHIN
      RETURN
```

```
      SUBROUTINE DPHIDH(A,B,H,HP,HPP,PHI,DPHI,DDPHI,X,RXY,R,MU,DRDT,
     1 DDRDT)
      DIMENSION X(1),DRDT(1),DDRDT(1)
      REAL LAMDA,MU
C
C         THE PROGRAM COMPUTES THE FIRST AND SECOND DERIVATIVES OF PHI AND H.
C
C         RXY = (X*X + Y*Y)**0.5
C         R = (X*X + Y*Y + Z*Z)**0.5
C         DRXY = D(RXY)/DT
C         DDRXY = D2(RXY)/DT2
C
      DRXY=(X(1)*DRDT(1) + X(2)*DRDT(2))/RXY
      W=X(1)*DRDT(2) - X(2)*DRDT(1)
      DDRXY=W*W/(RXY*RXY*RXY) + (X(1)*DDRDT(1) + X(2)*DDRDT(2))/RXY
      DZ=DRDT(3)
      DDZ=DDRDT(3)
C
      C=COS(PHI)
      S=SIN(PHI)
      ANG=2.*PHI
C
C         COMPUTE DPHI/DT AND DH/DT
C
      TERM=A*A*C*C + B*B*S*S
      TERM2=A*A/SQRT(TERM)
      ETA=TERM2 + H
      ETA2=B*B*TERM2/A/A + H
      CON=A*A - B*B
      Q=A*A*CON*SIN(ANG)/2./(TERM**1.5)
      A11=Q*C - ETA*S
      A12=C
      A21=B*B*Q*S/A/A + ETA2*C
      A22=S
      C1=DRXY
      C2=DZ
      DELT=A11*A22 - A12*A21
C
      DPHI=(A22*C1 - A12*C2)/DELT
      HP=(-A21*C1 + A11*C2)/DELT
C
C
C         COMPUTE D2(PHI)/DT2 AND D2H/DT2
C
      DQ=A*A*CON*(COS(ANG)/(TERM**1.5) + 3.*CON*SIN(ANG)*SIN(ANG)/4./
     1(TERM**2.5))*DPHI
      DETA=Q*DPHI + HP
      DETA2=B*B*Q*DPHI/A/A + HP
      B11=Q*C - ETA*S
      B12=C
      B21=B*B*Q*S/A/A + ETA2*C
      B22=S
      D1=DDRXY - DQ*DPHI*C + 2.*DETA*S*DPHI + ETA*C*DPHI*DPHI
C
      D2=DDZ - B*B*DQ*DPHI*S/A/A -2.*DETA2*C*DPHI + ETA2*S*DPHI*DPHI
C
      DELT2=B11*B22 - B12*B21
C
C
      DDPHI=(B22*D1 - B12*D2)/DELT2
      HPP=(-B21*D1 + B11*D2)/DELT2
C
C
      RETURN
      END
```

BEST AVAILABLE COPY

```
      SUBROUTINE VECPRD(A,B,C,NU)
      DIMENSION A(1),B(1)
      IF(NU.EQ.0)GO TO 5
C         THIS PROGRAM FORMS THE INNER PRODUCT OF TWO VECTORS.
      C=0.
      DO 1 I=1,3
    1 C=C + A(I)*B(I)
      RETURN
    5 C=0.
      DO 2 I=1,3
    2 C=C + A(I)*A(I)
      RETURN
      END
```

```
      SUBROUTINE SCANNER(T,R1,R2,R3,X1,X2,X3,N1,N2,N3,H,DR1,DR2,DR3,
     1                   C1,C2,C3)
C*****************************************************************************
C
C     THIS PROGRAM COMPUTES A LINEAR CORRECTION TO THE TIME WHEN THE
C     SCAN RATE CAN NOT BE CONSIDERED INFINITE
C
C     T                TIME AS FOUND IN NS ASSUMING INFINITE SCAN SPEED.
C     H                HEIGHT OF SATELLITE ABOVE EARTHS SURFACE
C     R1,R2,R3         MAPPING PT.
C     X1,X2,X3         EARTH FIXED SATELLITE POSITION
C     N1,N2,N3         NORMAL VECTOR TO EARTHS SURFACE
C     DR1,DR2,DR3      DISPLACEMENT OF MAPPING PT. FROM SATELLITE G.T..
C
C*****************************************************************************
      REAL N1,N2,N3
      S1=R1 - X1
      S2=R2 - X2
      S3=R3 - X3
      S=SQRT(S1*S1 + S2*S2 + S3*S3)
C
      SDOTH=-H*(S1*N1 + S2*N2 + S3*N3)
C
      DRDOTC=DR1*C1 + DR2*C2 + DR3*C3
C
      IF(DRDOTC.EQ.0) GO TO 5
      SIGN=DRDOTC/ABS(DRDOTC)
    5 IF(DRDOTC.EQ.0)SIGN=1.
      TERM=SDOTH/(S*H)
      E=ACOS(TERM)*SIGN
      DT=0.018355
      EMAX=0.10088
      DT=DT*(E/EMAX)
C
      T=T + DT
      RETURN
      END
```

BEST AVAILABLE COPY

The following are the program output for six cases and four sub-cases;

Case #	Description
1.0	Satellite groundtrack coordinates and Fourier series coefficients: (x_g, y_g) = map plane coordinates (ϕ,λ) = ellipsoid coordinates
*2.1	Forward transformation: $(\phi,\lambda) \rightarrow (x,y)$ for ϕ = 1.155860 radians λ = 2.375706 radians
*2.2	Forward transformation: for ϕ = -.002888 radians λ = -.122576 radians
*3.1	Inverse transformation: $(x,y) \rightarrow (\phi,\lambda)$ for x = 10099.660 km y = -59.587 km
*3.2	Inverse transformation: for x = 37787.170 km y = 791.363 km
*4.0	Distortion error analysis
**5.1	Forward transformation: (ϕ,λ) = (1.155869, 2.375706)
**5.2	Forward transformation: (ϕ,λ) = (-.002888, -.122576)
**6.1	Inverse transformation: (x,y) = 10099.59, -59.58)
**6.2	Inverse transformation: (x,y) = 37787.67, 791.390)

---

*Assumes instantaneous scanner sweep.

**Assumes constant scanner sweep rate of 314.9 deg/sec (13.62 Hz).

SPACE OBLIQUE MERCATOR

SATELLITE GROUND TRACK PROJECTION

CASE 1.0

INPUT PARAMETERS DEFINING THE CURRENT RUN.

POSITION	VELOCITY
0.00000	7.39381
1140.60150	0.00000
7201.47443	0.00000

ORBITAL ELEMENTS

7291.24171	SEMI-MAJOR AXIS OF THE ORBIT
.00000	ECCENTRICITY OF THE ORBIT
1.72788	INCLINATION OF THE ORBIT
3.14159	LONGITUDE OF THE ASCENDING NODE
1.57080	ARGUMENT OF PERIFOCUS
0.00000	TIME OF PERIFOCAL PASSAGE
6196.02000	SATELLITE ORBITAL PERIOD

EARTH RELATED PARAMETERS

FLATTENING FACTOR=	.33523E-02
SEMI-MAJOR=	.63782E+04
SEMI-MINOR AXIS=	.63568E+04

TABLE D1

SPACE OBLIQUE MERCATOR

SATELLITE GROUND TRACK PROJECTION

CASE 1.0

INITIAL TIME= 0.0000
FINAL TIME= 6196.0200

INTEGRATION STEP SIZE = (SATELLITE PERIOD)/400. = 15.4901 SEC.

RESULTS OF GROUND TRACK INTEGRATION FOR THE TIME INCREMENT DELTA = (PERIOD)/100.

XG	YG	F	PHI	LAMDA	TIME
0.	.9163546E+03	0.	.1414619E+01	.1570796E+01	0.0000
.4042696E+03	.9145479E+03	-.8935153E-02	.1402682E+01	.1183895E+01	61.9602
.8085198E+03	.9091349E+03	-.1783516E-01	.1371102E+01	.8824294E+00	123.9204
.1212732E+04	.9001368E+03	-.2666500E-01	.1327403E+01	.6732978E+00	185.8806
.1616887E+04	.8875888E+03	-.3538993E-01	.1276930E+01	.5291364E+00	247.8408
.2020969E+04	.8715401E+03	-.4397561E-01	.1222606E+01	.4261114E+00	309.8010
.2424959E+04	.8520536E+03	-.5238822E-01	.1165965E+01	.3491731E+00	371.7612
.2828845E+04	.8292057E+03	-.6059463E-01	.1107843E+01	.2893190E+00	433.7214
.3232612E+04	.8030862E+03	-.6856249E-01	.1048721E+01	.2410802E+00	495.6816
.3636250E+04	.7737973E+03	-.7626037E-01	.9888910E+00	.2010192E+00	557.6418
.4039748E+04	.7414543E+03	-.8365788E-01	.9285391E+00	.1668938E+00	619.6020
.4443100E+04	.7061842E+03	-.9072580E-01	.8677883E+00	.1371898E+00	681.5622
.4846302E+04	.6681256E+03	-.9743620E-01	.8067235E+00	.1108515E+00	743.5224
.5249350E+04	.6274283E+03	-.1037625E+00	.7454054E+00	.8712177E-01	805.4826
.5652245E+04	.5842525E+03	-.1096797E+00	.6838790E+00	.6544361E-01	867.4428
.6054989E+04	.5387682E+03	-.1151643E+00	.6221785E+00	.4539751E-01	929.4030
.6457587E+04	.4911548E+03	-.1201947E+00	.5603311E+00	.2666074E-01	991.3632
.6860046E+04	.4416000E+03	-.1247507E+00	.4983591E+00	.8979925E-02	1053.3234
.7262376E+04	.3902993E+03	-.1288144E+00	.4362812E+00	-.7847834E-02	1115.2836
.7664588E+04	.3374554E+03	-.1323695E+00	.3741139E+00	-.2398821E-01	1177.2438
.8066694E+04	.2832770E+03	-.1354020E+00	.3118720E+00	-.3957926E-01	1239.2040
.8468711E+04	.2279782E+03	-.1378997E+00	.2495692E+00	-.5473844E-01	1301.1642
.8870654E+04	.1717775E+03	-.1398528E+00	.1872184E+00	-.6956799E-01	1363.1244
.9272540E+04	.1148973E+03	-.1412534E+00	.1248319E+00	-.8415912E-01	1425.0846
.9674388E+04	.5756252E+02	-.1420961E+00	.6242184E-01	-.9859532E-01	1487.0448
.1007622E+05	-.2501425E-08	-.1423773E+00	.6353353E-13	-.1129552E+00	1549.0050
.1047805E+05	-.5756252E+02	-.1420961E+00	-.6242184E-01	-.1273152E+00	1610.9652
.1087989E+05	-.1148973E+03	-.1412534E+00	-.1248319E+00	-.1417514E+00	1672.9254
.1128178E+05	-.1717775E+03	-.1398528E+00	-.1872184E+00	-.1563425E+00	1734.8856
.1168372E+05	-.2279782E+03	-.1378997E+00	-.2495692E+00	-.1711720E+00	1796.8458
.1208574E+05	-.2832770E+03	-.1354020E+00	-.3118720E+00	-.1863312E+00	1858.8060
.1248785E+05	-.3374554E+03	-.1323695E+00	-.3741139E+00	-.2019223E+00	1920.7662
.1289006E+05	-.3902993E+03	-.1288144E+00	-.4362812E+00	-.2180626E+00	1982.7264
.1329239E+05	-.4416000E+03	-.1247507E+00	-.4983591E+00	-.2348904E+00	2044.6866
.1369485E+05	-.4911548E+03	-.1201947E+00	-.5603311E+00	-.2525712E+00	2106.6468
.1409745E+05	-.5387682E+03	-.1151643E+00	-.6221785E+00	-.2713080E+00	2168.6070
.1450019E+05	-.5842525E+03	-.1096797E+00	-.6838790E+00	-.2913541E+00	2230.5672
.1490308E+05	-.6274283E+03	-.1037625E+00	-.7454054E+00	-.3130323E+00	2292.5274
.1530613E+05	-.6681256E+03	-.9743620E-01	-.8067235E+00	-.3367619E+00	2354.4876
.1570933E+05	-.7061842E+03	-.9072580E-01	-.8677883E+00	-.3631002E+00	2416.4478
.1611269E+05	-.7414543E+03	-.8365788E-01	-.9285391E+00	-.3928042E+00	2478.4080
.1651618E+05	-.7737973E+03	-.7626037E-01	-.9888910E+00	-.4269296E+00	2540.3682
.1691982E+05	-.8030862E+03	-.6856249E-01	-.1048721E+01	-.4669907E+00	2602.3284
.1732359E+05	-.8292057E+03	-.6059463E-01	-.1107843E+01	-.5152295E+00	2664.2886
.1772747E+05	-.8520536E+03	-.5238822E-01	-.1165965E+01	-.5750836E+00	2726.2488
.1813147E+05	-.8715401E+03	-.4397561E-01	-.1222606E+01	-.6520219E+00	2788.2090
.1853555E+05	-.8875888E+03	-.3538993E-01	-.1276930E+01	-.7550469E+00	2850.1692
.1893970E+05	-.9001368E+03	-.2666500E-01	-.1327403E+01	-.8992083E+00	2912.1294
.1934391E+05	-.9091349E+03	-.1783516E-01	-.1371102E+01	-.1108340E+01	2974.0896
.1974816E+05	-.9145479E+03	-.8935153E-02	-.1402682E+01	-.1409805E+01	3036.0498
.2015243E+05	-.9163546E+03	.1193905E-12	-.1414619E+01	-.1796707E+01	3098.0100
.2055670E+05	-.9145479E+03	.8935153E-02	-.1402682E+01	-.2183608E+01	3159.9702
.2096095E+05	-.9091349E+03	.1783516E-01	-.1371102E+01	-.2485074E+01	3221.9304
.2136517E+05	-.9001368E+03	.2666500E-01	-.1327403E+01	-.2694205E+01	3283.8906

.2176932E+05	-.8875888E+03	.3538993E-01	-.1276930E+01	-.2838367E+01	3345.8508
.2217340E+05	-.8715401E+03	.4397561E-01	-.1222606E+01	-.2941392E+01	3407.8110
.2257739E+05	-.8520536E+03	.5238822E-01	-.1165965E+01	-.3018330E+01	3469.7712
.2298128E+05	-.8292057E+03	.6059463E-01	-.1107843E+01	-.3078184E+01	3531.7314
.2338505E+05	-.8030862E+03	.6856249E-01	-.1048721E+01	-.3126423E+01	3593.6916
.2378868E+05	-.7737973E+03	.7626037E-01	-.9888910E+00	.3116701E+01	3655.6518
.2419218E+05	-.7414543E+03	.8365788E-01	-.9285391E+00	.3082576E+01	3717.6120
.2459553E+05	-.7061842E+03	.9072580E-01	-.8677883E+00	.3052872E+01	3779.5722
.2499874E+05	-.6681256E+03	.9743620E-01	-.8067235E+00	.3026534E+01	3841.5324
.2540178E+05	-.6274283E+03	.1037625E+00	-.7454054E+00	.3002804E+01	3903.4926
.2580468E+05	-.5842525E+03	.1096797E+00	-.6838790E+00	.2981126E+01	3965.4528
.2620742E+05	-.5387682E+03	.1151643E+00	-.6221785E+00	.2961080E+01	4027.4130
.2661002E+05	-.4911548E+03	.1201947E+00	-.5603311E+00	.2942343E+01	4089.3732
.2701248E+05	-.4416000E+03	.1247507E+00	-.4983591E+00	.2924662E+01	4151.3334
.2741481E+05	-.3902993E+03	.1288144E+00	-.4362812E+00	.2907834E+01	4213.2936
.2781702E+05	-.3374554E+03	.1323695E+00	-.3741139E+00	.2891694E+01	4275.2538
.2821913E+05	-.2832770E+03	.1354020E+00	-.3118720E+00	.2876103E+01	4337.2140
.2862114E+05	-.2279782E+03	.1378997E+00	-.2495692E+00	.2860944E+01	4399.1742
.2902309E+05	-.1717775E+03	.1398528E+00	-.1872184E+00	.2846114E+01	4461.1344
.2942497E+05	-.1148973E+03	.1412534E+00	-.1248319E+00	.2831523E+01	4523.0946
.2982682E+05	-.5756252E+02	.1420961E+00	-.6242184E-01	.2817087E+01	4585.0548
.3022865E+05	.1889153E-08	.1423773E+00	-.7765209E-12	.2802727E+01	4647.0150
.3063048E+05	.5756252E+02	.1420961E+00	.6242184E-01	.2788367E+01	4708.9752
.3103233E+05	.1148973E+03	.1412534E+00	.1248319E+00	.2773931E+01	4770.9354
.3143421E+05	.1717775E+03	.1398528E+00	.1872184E+00	.2759340E+01	4832.8956
.3183616E+05	.2279782E+03	.1378997E+00	.2495692E+00	.2744510E+01	4894.8558
.3223817E+05	.2832770E+03	.1354020E+00	.3118720E+00	.2729351E+01	4956.8160
.3264028E+05	.3374554E+03	.1323695E+00	.3741139E+00	.2713760E+01	5018.7762
.3304249E+05	.3902993E+03	.1288144E+00	.4362812E+00	.2697620E+01	5080.7364
.3344482E+05	.4416000E+03	.1247507E+00	.4983591E+00	.2680792E+01	5142.6966
.3384728E+05	.4911548E+03	.1201947E+00	.5603311E+00	.2663111E+01	5204.6568
.3424988E+05	.5387682E+03	.1151643E+00	.6221785E+00	.2644374E+01	5266.6170
.3465262E+05	.5842525E+03	.1096797E+00	.6838790E+00	.2624328E+01	5328.5772
.3505552E+05	.6274283E+03	.1037625E+00	.7454054E+00	.2602650E+01	5390.5374
.3545857E+05	.6681256E+03	.9743620E-01	.8067235E+00	.2578920E+01	5452.4976
.3586177E+05	.7061842E+03	.9072580E-01	.8677883E+00	.2552582E+01	5514.4578
.3626512E+05	.7414543E+03	.8365788E-01	.9285391E+00	.2522878E+01	5576.4180
.3666862E+05	.7737973E+03	.7626037E-01	.9888910E+00	.2488753E+01	5638.3782
.3707226E+05	.8030862E+03	.6856249E-01	.1048721E+01	.2448691E+01	5700.3384
.3747602E+05	.8292057E+03	.6059463E-01	.1107843E+01	.2400453E+01	5762.2986
.3787991E+05	.8520536E+03	.5238822E-01	.1165965E+01	.2340599E+01	5824.2588
.3828390E+05	.8715401E+03	.4397561E-01	.1222606E+01	.2263660E+01	5886.2190
.3868798E+05	.8875888E+03	.3538993E-01	.1276930E+01	.2160635E+01	5948.1792
.3909214E+05	.9001368E+03	.2666500E-01	.1327403E+01	.2016474E+01	6010.1394
.3949635E+05	.9091349E+03	.1783516E-01	.1371102E+01	.1807342E+01	6072.0996
.3990060E+05	.9145479E+03	.8935153E-02	.1402682E+01	.1505877E+01	6134.0598
.4030487E+05	.9163546E+03	.2174229E-12	.1414619E+01	.1118975E+01	6196.0200

COEFFICIENTS FOR THE FOURIER FIT XG,YG AND F

X-COEFFS.	Y-COEFFS.	F-COEFFS.
.6504961E+01	.9163549E+03	.1100839E-12
.4082919E-08	-.2364052E-08	-.1423582E+00
-.6160920E-09	.9164511E+03	.2427713E-12
-.1907859E-10	.3234030E-08	.3637487E-14
.9736479E+01	-.7500633E-10	.2380546E-12
.3152180E-09	.2233568E-08	.1912630E-04
.1827269E-09	-.9647972E-01	.2916712E-12
-.1715543E-10	.2090698E-08	-.9321191E-15
.2320499E-02	-.3590699E-10	.3577839E-12
-.1112406E-09	.2032961E-08	.3275758E-07

BEST AVAILABLE COPY

SPACE OBLIQUE MERCATOR

FORWARD TRANSFORMATION

CASE 2.1

FORWARD TRANSFORMATION TO FIND X,Y GIVEN PHI,LAMDA

```
PHI   =    1.155860
LAMDA =    2.375706
```

X,Y,Z EARTH-FIXED ELLIPSOIDIAL COORDINATES

```
X=-1858.483932
Y= 1787.339436
Z= 5814.176682
```

SCAN VECTOR IN TC PLANE AT TIME T-STAR

```
 .4058020E+00
 .9006060E+00
-.1556712E+00
```

NORMAL VECTOR N AT TIME T-STAR

```
-.2870451E+00
 .2872943E+00
 .9138201E+00
```

DISPLACEMENT VECTOR DELTA-R AT TIME T-STAR

```
-.2252466E+02
-.5021357E+02
 .8445968E+01
```

T-STAR= 5809.886613

MAP PROJECTION COORDINATES

```
X =       .3778774E+05 Y =       .7913939E+03
```

SPACE OBLIQUE MERCATOR

FORWARD TRANSFORMATION

CASE 2.2

FORWARD TRANSFORMATION TO FIND X,Y GIVEN PHI,LAMDA

PHI = -.002888
LAMDA = -.122576

X,Y,Z EARTH-FIXED ELLIPSOIDIAL COORDINATES

X= 6330.283038
Y= -779.851459
Z= -18.299651

SCAN VECTOR IN TC PLANE AT TIME T-STAR

.1116266E+00
.9813602E+00
-.1564344E+00

NORMAL VECTOR N AT TIME T-STAR

.9935073E+00
-.1136884E+00
-.4266305E-02

DISPLACEMENT VECTOR DELTA-R AT TIME T-STAR

-.6470795E+01
-.5472793E+02
.8729411E+01

T-STAR= 1553.239634

MAP PROJECTION COORDINATES

X = .1009966E+05 Y = -.5958681E+02

SPACE OBLIQUE MERCATOR

INVERSE TRANSFORMATION

CASE 3.1

INVERSE TRANSFORMATION TO FIND PHI, LAMDA GIVEN X,Y

X= .1009966E+05 Y= -.5958680E+02

VECTOR DELTA-R MAP PROJECTION PLANE COMPONENTS AT TIME T-STAR

X-COMPONENT = -.4017802E+01
Y-COMPONENT = -.5565014E+02

NORMAL VECTOR N AT TIME T-STAR

.9935073E+00
-.1136884E+00
-.4266305E-02

VECTOR T-PRIME MAP PROJECTION PLANE COMPONENTS AT TIME T-STAR

X-COMPONENT = -.9974039E+00
Y-COMPONENT = .7201009E-01

FINAL PARTIALS USED IN THE PHI, LAMDA INVERSION

DX/D(PHI) = -.6312812E+04
DX/D(LAMDA) = -.5418130E+03
DY/D(PHI) = -.5379900E+03
DY/D(LAMDA) = .6354618E+04

THE (PHI,LAMDA) OF THE MAP PLANE POINT (X,Y)

PHI = -.2888447E-02
LAMDA = -.1225762E+00

T-STAR= 1553.239635

SPACE OBLIQUE MERCATOR

INVERSE TRANSFORMATION

CASE 3.2

```
INVERSE TRANSFORMATION TO FIND PHI, LAMDA GIVEN X,Y

X=      .3778717E+05 Y=      .7913626E+03

VECTOR DELTA-R MAP PROJECTION PLANE COMPONENTS AT TIME T-STAR

          X-COMPONENT =      .1529013E+01
          Y-COMPONENT =     -.5565628E+02

          NORMAL VECTOR N AT TIME T-STAR

                -.2871243E+00
                 .2873215E+00
                 .9137866E+00

VECTOR T-PRIME MAP PROJECTION PLANE COMPONENTS AT TIME T-STAR

          X-COMPONENT =     -.9996228E+00
          Y-COMPONENT =     -.2746207E-01

     FINAL PARTIALS USED IN THE PHI, LAMDA INVERSION

          DX/D(PHI)   =      .6007216E+04
          DX/D(LAMDA) =     -.8785523E+03
          DY/D(PHI)   =     -.2176227E+04
          DY/D(LAMDA) =     -.2424541E+04

THE (PHI,LAMDA) OF THE MAP PLANE POINT (X,Y)

          PHI   =     .1155777E+01
          LAMDA =     .2375793E+01

          T-STAR= 5809.798322
```

BEST AVAILABLE COPY

SPACE OBLIQUE MERCATOR

DISTORTION ERROR ANALYSIS*

CASE 4.0

LENGTH DISTORTIONS FOR PTS. SYMMETRICALLY PLACED ON BOTH SIDES OF THE SATELLITE GROUND TRACK FOR THE DISPLACEMENT INCREMENT DELTA = 55.66 KM.

PHI,LAMDA OF THE GROUND TRACK 81.05172 90.00000

TIME ALONG THE SATELLITE GROUND TRACK = 0.00000

$\frac{\partial s'}{\partial s}\Big\|_\lambda$	$\frac{\partial s'}{\partial s}\Big\|_\phi$	
.998428	1.000498	
.998941	1.000219	
.999299	1.000177	
.999605	1.000114	
.999834	1.000052	(SATELLITE GROUND TRACK)
.999938	1.000031	
.999697	1.000151	
.999320	1.000340	
.998791	1.000603	

PHI,LAMDA OF THE GROUND TRACK 44.46685 5.65505

TIME ALONG THE SATELLITE GROUND TRACK = 774.50250

1.000538	.998860	
1.000299	.999362	
1.000131	.999718	
1.000032	.999930	
1.000000	1.000000	(SATELLITE GROUND TRACK)
1.000031	.999931	
1.000124	.999725	
1.000275	.999385	
1.000480	.998915	

*The first (and last) point at ϕ=81.05° indicates a slight numerical difficulty in that small distortions are present on the ground track; whereas zero distortion is present on the ground track elsewhere, including ϕ=-81.05°. We suspect that errors in the Fourier series approximation (being worse at either end of the fit) are the culprit here. It has also been observed that these partials indicate small but significant loss of conformality (which is counter intuitive). This is a point which should receive careful attention in future work on this problem.

PHI,LAMDA OF THE GROUND TRACK -.09793 -6.49437

TIME ALONG THE SATELLITE GROUND TRACK = 1550.70146

1.000558	.998839	
1.000314	.999346	
1.000139	.999709	
1.000035	.999927	
1.000000	1.000000	(SATELLITE GROUND TRACK)
1.000035	.999927	
1.000140	.999709	
1.000314	.999346	
1.000558	.998839	

PHI,LAMDA OF THE GROUND TRACK -44.69872 -18.68870

TIME ALONG THE SATELLITE GROUND TRACK = 2327.59702

1.000479	.998916	
1.000274	.999386	
1.000124	.999725	
1.000031	.999931	
1.000000	1.000000	(SATELLITE GROUND TRACK)
1.000032	.999930	
1.000131	.999718	
1.000299	.999362	
1.000538	.998861	

PHI,LAMDA OF THE GROUND TRACK -81.04555 -105.11219

TIME ALONG THE SATELLITE GROUND TRACK = 3103.78611

.998793	1.000602	
.999321	1.000339	
.999698	1.000151	
.999924	1.000038	
1.000000	1.000000	(SATELLITE GROUND TRACK)
.999925	1.000038	
.999698	1.000151	
.999322	1.000339	
.998796	1.000603	

PHI,LAMDA OF THE GROUND TRACK -44.14007 172.58558

TIME ALONG THE SATELLITE GROUND TRACK = 3878.27456

1.000482	.998914	
1.000275	.999384	
1.000124	.999724	
1.000032	.999931	
1.000000	1.000000	(SATELLITE GROUND TRACK)
1.000032	.999930	
1.000131	.999718	
1.000299	.999361	
1.000539	.998860	

PHI,LAMDA OF THE GROUND TRACK .23428 160.53058

TIME ALONG THE SATELLITE GROUND TRACK = 4651.07361

1.000558	.998839	
1.000314	.999346	
1.000140	.999709	
1.000035	.999927	
1.000000	1.000000	(SATELLITE GROUND TRACK)
1.000035	.999927	
1.000139	.999709	
1.000314	.999346	
1.000558	.998839	

PHI,LAMDA OF THE GROUND TRACK 44.56129 148.42096

TIME ALONG THE SATELLITE GROUND TRACK = 5423.18305

1.000538	.998860	
1.000299	.999362	
1.000131	.999718	
1.000032	.999930	
1.000000	1.000000	(SATELLITE GROUND TRACK)
1.000031	.999931	
1.000124	.999725	
1.000274	.999385	
1.000480	.998915	

PHI,LAMDA OF THE GROUND TRACK 81.05172 64.12310

TIME ALONG THE SATELLITE GROUND TRACK = 6195.99195

.998429	1.000498	
.998941	1.000219	
.999299	1.000177	
.999605	1.000114	
.999834	1.000052	(SATELLITE GROUND TRACK)
.999969	1.000010	
.999996	1.000001	
.999903	1.000036	
.999684	1.000121	

SPACE OBLIQUE MERCATOR

FORWARD TRANSFORMATION

CASE 5.1

FORWARD TRANSFORMATION TO FIND X,Y GIVEN PHI,LAMDA

PHI = 1.155860
LAMDA = 2.375706

X,Y,Z EARTH-FIXED ELLIPSOIDIAL COORDINATES

X=-1858.483932
Y= 1787.339436
Z= 5814.176682

SCAN VECTOR IN TC PLANE AT TIME T-STAR

.4058020E+00
.9006060E+00
-.1556712E+00

NORMAL VECTOR N AT TIME T-STAR

-.2870451E+00
.2872943E+00
.9138201E+00

DISPLACEMENT VECTOR DELTA-R AT TIME T-STAR

-.2252466E+02
-.5021357E+02
.8445968E+01

T-STAR= 5809.875747

MAP PROJECTION COORDINATES

X = .3778767E+05 Y = .7913901E+03

SPACE OBLIQUE MERCATOR

FORWARD TRANSFORMATION

CASE 5.2

FORWARD TRANSFORMATION TO FIND X,Y GIVEN PHI,LAMDA

```
PHI   =    -.002888
LAMDA =    -.122576
```

X,Y,Z EARTH-FIXED ELLIPSOIDIAL COORDINATES

```
X= 6330.283038
Y= -779.851459
Z=  -18.299651
```

SCAN VECTOR IN IC PLANE AT TIME T-STAR

```
 .1116266E+00
 .9813602E+00
-.1564344E+00
```

NORMAL VECTOR N AT TIME T-STAR

```
 .9935073E+00
-.1136884E+00
-.4266305E-02
```

DISPLACEMENT VECTOR DELTA-R AT TIME T-STAR

```
-.6470795E+01
-.5472793E+02
 .8729411E+01
```

T-STAR= 1553.228533

MAP PROJECTION COORDINATES

```
X =     .1009959E+05 Y =     -.5957649E+02
```

BEST AVAILABLE COPY

SPACE OBLIQUE MERCATOR

INVERSE TRANSFORMATION

CASE 6.1

INVERSE TRANSFORMATION TO FIND PHI, LAMDA GIVEN X,Y

```
X=      .1009959E+05 Y=      -.5957649E+02

VECTOR DELTA-R MAP PROJECTION PLANE COMPONENTS AT TIME T-STAR

          X-COMPONENT =      -.4017801E+01
          Y-COMPONENT =      -.5565011E+02

          NORMAL VECTOR N AT TIME T-STAR

                .9935080E+00
               -.1136833E+00
               -.4243970E-02

VECTOR T-PRIME MAP PROJECTION PLANE COMPONENTS AT TIME T-STAR

          X-COMPONENT =      -.9974039E+00
          Y-COMPONENT =       .7201010E-01

       FINAL PARTIALS USED IN THE PHI, LAMDA INVERSION

          DX/D(PHI)   =      -.6312812E+04
          DX/D(LAMDA) =      -.5418128E+03
          DY/D(PHI)   =      -.5379899E+03
          DY/D(LAMDA) =       .6354618E+04

     THE (PHI,LAMDA) OF THE MAP PLANE POINT (X,Y)

          PHI   =      -.2888484E-02
          LAMDA =      -.1225762E+00

          T-STAR= 1553.217466
```

SPACE OBLIQUE MERCATOR

INVERSE TRANSFORMATION

CASE 6.2

INVERSE TRANSFORMATTON TO FTND PHI, LAMDA GIVEN X,Y

```
X=      .3778767E+05 Y=      .7913901E+03

VECTOR DELTA-R MAP PROJECTION PLANE COMPONENTS AT TIME T-STAR

      X-COMPONENT =       .1528720E+01
      Y-COMPONENT =      -.5565614E+02

      NORMAL VECTOR N AT TIME T-STAR

           -.2870649E+00
            .2873011E+00
            .9138117E+00

VECTOR T-PRIME MAP PROJECTION PLANE COMPONENTS AT TIME T-STAR

      X-COMPONENT =      -.9996230E+00
      Y-COMPONENT =      -.2745687E-01

   FINAL PARTIALS USED IN THE PHI, LAMDA INVERSION

      DX/D(PHI)    =       .6007062E+04
      DX/D(LAMDA)  =      -.8785653E+03
      DY/D(PHI)    =      -.2176685E+04
      DY/D(LAMDA)  =      -.2424019E+04

   THE (PHI,LAMDA) OF THE MAP PLANE POINT (X,Y)

      PHI   =      .1155860E+01
      LAMDA =      .2375706E+01

      T-STAR= 5809.864590
```

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ETL-0131	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Formulation of a Space Oblique Mercator Map Projection		5. TYPE OF REPORT & PERIOD COVERED Contract Report
		6. PERFORMING ORG. REPORT NUMBER UVA/525023/~~[illegible]~~ ESS 77/105
7. AUTHOR(s) John L. Junkins James D. Turner		8. CONTRACT OR GRANT NUMBER(s) DAAG53-76-C-0067
9. PERFORMING ORGANIZATION NAME AND ADDRESS Research Laboratories for the Engineering Sciences University of Virginia Charlottesville, Virginia 22901		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE November 1977
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

remote sensing, LANDSAT, ERTS, conformal, map projection, Mercator.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report develops a *dynamic* map projection especially suited of processing and displaying of satellite electro-optical remote sensing of the earth's surface. The new map projection (the Space Oblique Mercator) projects the satellite groundtrack from the ellipsoid into the map plane, free of length distortion and free of *normal view curvature* distortion. The length and curvature distortions in the finite sensed region are negligible for most applications. The report details the formulation, provides numerical examples for the LANDSAT-1 multi-spectral scanner, and includes FORTRAN IV software as an appendix.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DISTRIBUTION LIST

Copy No.	
1 - 50	Computer Science Laboratory Code 82000 U.S. Army Engineer Topographic Laboratories Fort Belvoir, Virginia 22060 Attn: Mr. L. A. Gambino
51 - 100	United States Department of the Interior Geological Survey Reston, Virginia 22092 Attn: Dr. A. P. Colvocoresses
101 - 110	J. L. Junkins
111 - 115	J. D. Turner
116	L. S. Fletcher
117 - 118	E. H. Pancake Clark Hall
119	I. A. Fischer Office of Sponsored Programs
120	RLES Files